Tech:Residential

From GridLAB-D Wiki

House

Thermal loads are implemented using the ETP model, which determines the heat flow between an inner and outer air mass and one solid mass embedded within the inner air mass.

The basic etp parameters are

- Ua: the conductance between the inner and outer air mass
- Ca: the heat capacity of the internal air mass
- Cm: the heat capacity of the internal solid mass
- Um: the conductance between the inner air mass and the inner solid mass
- To: the outer air temperature
- Ti: the inner air temperature
- Tm: the inner mass temperature
- Qa: the heat flux to the interior air mass
- Qm: the heat flux to the interior solid mass (this is not supported yet)

The solution to the ETP model is based on the two ordinary differential equations (ODEs)

Contents

$$\begin{aligned}\frac{dT_i}{dt} &= \frac{1}{C_a}\left[T_m U_m - T_i\left(U_a + U_m\right) + Q_a + T_0 U_a\right]\\ \frac{dT_m}{dt} &= \frac{1}{C_m}\left[U_m\left(T_i - T_m\right) + Q_m\right]\end{aligned}$$

The general first-order ODEs are found by inspection:

$$\begin{aligned}\dot{T}_i &= c_1 T_i + c_2 T_m + c_3\\ \dot{T}_m &= c_4 T_i + c_5 T_m + c_6\end{aligned}$$

where the constants c_1 through c_6 are

- $c_1 = -\frac{U_a + U_m}{C_a}$
- $c_2 = \frac{U_m}{C_a}$
- $c_3 = \frac{Q_a + U_a T_0}{C_a}$
- $c_4 = \frac{U_m}{C_m}$
- $c_5 = -\frac{U_m}{C_m}$
- $c_6 = \frac{Q_m}{C_m}$

The general form of the second-order ODE is given by

$$p_1 \ddot{T}_i + p_2 \dot{T}_i + p_3 T_i + p_4 = 0$$

where

- $p_1 = \frac{1}{c_2}$
- $p_2 = -\frac{c_1 + c_5}{c_2}$
- $p_3 = \frac{c_1 c_5}{c_2} - c_4$
- $p_4 = \frac{c_3 c_5}{c_2} - c_6$

where

- r_1 and r_2 are the roots of $p_1 r^2 + p_2 r + p_3 = 0$,
- $k_1 = \frac{r_2 T_{i,0} + r_2 p_4 / p_3 - \dot{T}_{i,0}}{r_2 - r_1}$,
- $k_2 = \frac{T_{i,0} - r_1 k_1}{r_2}$
- t is the elapsed time,
- $T_{i,0}$ is the initial value of T_i, and
- $\dot{T}_{i,0} = c_1 T_{i,0} + c_2 T_{m,0} + (c_1 + c_2) T_0 + c_7$

with

- $c_7 = \frac{Q_a}{C_a}$, and
- $T_{m,t} = k_1 \frac{r_1 - c_1}{c_2} + k_2 \frac{r_2 - c_1}{c_2} - \frac{p_4}{p_3} + \frac{c_6}{c_2}$

Modeling Assumptions

The house model assumes that only envelope characteristics, solar gain through windows and internal gain contribute to the HVAC load. Only air conditioners and heat pumps are modeled in the current implementation.

Modeling Approach

The model approach that is used to estimate thermal loads is called an equivalent thermal parameter (ETP) modeling approach. Because the ETP approach has been proven to provide reasonably model the residential (and small commercial building) loads and energy consumption and also because it is based on first principles, this modeling approach has been chosen for the current work (Sonderegger 1978; Subbarao 1981; Wilson et al. 1985; Pratt and Taylor 1994). The electric circuit analog of an ETP model used to simulate heating and cooling loads in a typical residence is shown in Figure 3. The heat transfer properties are represented by equivalent electrical components with associated parameters for modeling the thermostatically controlled heating, ventilation, and air-conditioning (HVAC) system.

where,

- C_{air} – air heat capacity (Btu/^{o}F or J/^{o}C)
- C_{mass} – mass (of the building and its content) heat capacity (Btu/^{o}F or J/^{o}C)
- UA_{wall} – the gain/heat loss coefficient (Btu/^{o}F.h or W/^{o}C) to the ambient
- UA_{mass} – the gain/heat loss coefficient (Btu/^{o}F.h or W/^{o}C) between air and mass
- T_o – CLTD$_c$ + T$_{air}$ (^{o}F or ^{o}C)
- $T_{ambient}$ – ambient temperature (^{o}F or ^{o}C)
- T_{air} – air temperature inside the house (^{o}F or ^{o}C)

- T_{mass} – mass temperature inside the house (^{o}F or ^{o}C)
- Q_{HVAC} – heat rate for HVAC (Btu/hr or W)
- $Q_{internal}$ – heat rate from other appliance, plug loads, lights and people in the residence (Btu/h or W)
- Q_{solar} – heat gain from solar (Btu/h or W)

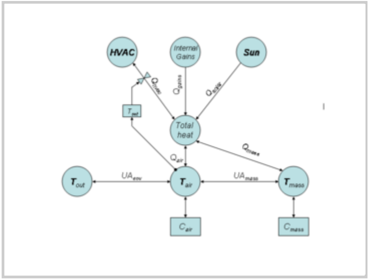

Figure 1 – ETP Representation of the Typical Residences

A state space description of the ETP model is:

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx + Du$$

where (**there's something important missing here: A, B, C, D are not described**)

- $R1 = 1/UA_{insul}$
- $R2 = 1/UA_{mass}$
- $Q = Q_{HVAC} + Q_{solar} + Q_{internal}$

Because there is wide diversity in the thermal parameters (UA, mass, efficiency of equipment, and over sizing factor) used to compute the load across homes, ranges can be provided for these parameters. Because of regional difference in construction practices and wide variation even within a region, a range of values can be assigned for critical parameters based on metering studies (Pratt et al. 1990). While estimating energy consumption of individual homes, the thermal parameters are randomly (from the established range using known distribution, for example, uniform) assigned to each home.

The internal gains in the home are computed external to the house object and are assumed to be inputs to the ETP model. These gains represent heat from other appliances (such as range, microwave), plug loads, lights, and people.

Solving the ETP model (simultaneously for Tair and Tmass), we can obtain the cooling/heating load of an individual home as a function of time. The energy consumption to meet the comfort needs in the home can then be computed by converting the thermal loads by using typical manufacturer-provided air conditioner part load performance data. Using the same simulation model, the energy consumption of a population of homes can also be computed. While simulating a population of homes, the energy consumption for each home is computed simultaneously at each time step. Therefore, we get an accurate representation of distribution feeder load when electric loads are aggregated.

A single ETP model with different thermal parameters can be used to represent all homes in the population for simplicity, or if there is a need, multiple ETP models with different thermal parameters can be used. Therefore, in addition to changing input parameters while simulating a population of homes, the ETP model can also be changed to accurately represent a given building stock. This ensures that it accurately reproduces the effects of load diversity.

Limitation and Future Improvements

The approach described in the previous section only accounts for sensible loads. This does not lead to a significant error in heating load calculation (because most heating load is sensible); it does lead to some error in the estimation of the cooling load.

End-uses

Generic enduses can be implemented using the **residential_enduse** object. This object requires a schedule and a loadshape definition (see Built in schedules and loadshapes). All other end-uses are (or will soon be) inheriting the properties and methods of the **residential_enduse** object.

Enduse Structure

Objects that need to use the *enduse* structure can either inherit the needed properties from an object that already includes one, e.g. *residential_enduse*, or include one of their own.

When the *enduse* structure is inherited, it is strongly recommend that it's name be set during object creation. Usually the name given to the enduse is the same of the class implementing the enduse. The name can be assigned during *create* by using the command

```
load.name = "lights";
```

However, when an object implements multiple enduses, then each one is embedded and the name of the property should be given when the end-use is published, i.e.,

```
...
PT_enduse, "lights", PADDR(enduse),
...
```

In addition, the corresponding shape must be defined and linked. When inheriting the enduse structure, this is already done. However, when defining the enduse structure in an object, it must done explicitly during *create* using the command

```
lights.shape = lighting_shape;
```

where *shape* is a property of type *loadshape* and can be published as

```
...
PT_loadshape, "shape", PADDR(lighting_shape),
...
```

It is recommended that shape for embedded enduses be published with the enduse name as a prefix, e.g.,

```
...
```

```
PT_loadshape, "lights.shape", PADDR(lighting_shape),
...
```

The following updates are made before any *presync* calls are made

shape->schedule
: the schedule is updated to adjust the schedule value according to the target time of the sync event. The value *next_t' is updated to give the time of the next expected schedule change.*

shape
: the shape is updated to make the state of the loadshape correspond to the schedule value. The value t2 is updated to correspond to the expected time of the next change in demand.

enduse
: this enduse energy is updated to include the power consumed up to the target time of the sync event.

In addition, during *sync* events, objects that use *enduse* properties must explicitly update their enduse structures using the *gl_enduse_sync*.

Important
: omitting *gl_enduse_sync* after *enduse* values are updated will result in erroneous output.

Water Heater

Typical residential electric water heaters range in size from about 30 gallons up to 120 gallons or so, with most tanks falling in the range of about 40 to 80 gallons. Hot water is drawn from the top of the tank to service the home's water loads, with cold make-up water injected near the bottom of the tank. When a hot water draw is in progress, the tank will tend to have a layer of cold water at the bottom.

Water is typically heated by two elements, one located near the bottom of the tank and one located higher in the tank, usually halfway to two-thirds of the way to the top. The lower element shoulders most of the heating load, with the upper element engaging only when the cold water layer has reached it. The two elements are controlled by independent thermostats, but only one is permitted to be on at a time, the upper element having priority. The design is intended to allow rapid (re)heating of the smaller volume of water above the top element so that full-temperature water is available as soon as possible after a depletion. Water heater elements range in capacity from about 1500 Watts to 6000 Watts, with 4500 Watts being common.

Thermostatic controls have a “dead band” associated with the setpoint, which prevents rapid cycling of power to the elements which would result if the turn-on temperature equaled the turn-off temperature. The dead band is typically a few degrees above and below the nominal setpoint.

Most heaters are shipped with both the upper and lower thermostats set to the same temperature (often 120 degrees-F), but they may be modified at installation or a later time. Some manufacturers recommend that if hotter water is desired, only the lower element be set to a higher temperature. This will maintain the entire tank at the higher temperature but allow for rapid recovery of the upper volume to the lower setting following a depletion. The net effect on energy use is negligible if the upper thermostat is set to a lower setting. However, some homeowners wrongly set the upper element to a higher temperature than the lower one, which results in somewhat different behavior. Because of thermal stratification, the upper volume will be maintained at a higher temperature than the lower volume, resulting in somewhat lower standby losses than when the entire tank is heated to the higher setting.

Modeling Assumptions

The GridLAB-D approach to modeling electric water heaters is designed to be computationally fast yet reasonably accurate. It accommodates the common two-element design and the possibility for "inverted" thermostat settings, wherein the upper element maintains a higher temperature than the lower element.

To achieve the necessary computational speed, we make the following assumptions:

1. Thermal stratification in the tank is not directly modeled. Depending on the situation, the water will be considered to be either of uniform temperature throughout the tank or "lumped" into two temperature regions (hot and cold layers).
2. The injection of cold inlet water at the bottom of the tank results in either complete mixing with the hot water in the tank or no mixing at all, depending on the volumetric flow rate.

The water heater simulation uses two very different models depending on the state of the tank at any given moment. The two models are:

1. One-Node Model – This is a simple, lumped-parameter electric analogue model that considers the entire tank to be a single "slug" of water at a uniform temperature. This model concerns the temperature of the water at any given time and/or the time required for the temperature to move between two specified points.
2. Two-Node Model – This model, which applies when the heater is in a state of partial depletion, considers the heater to consist of two slugs of water, each at a uniform temperature. The upper "hot" node is near the heater's setpoint temperature, while the lower "cold" node is near the inlet water temperature. This model concerns the location of the boundary between the hot and cold nodes, calculating the movement of that boundary as hot water is drawn from the tank and/or heat is added to the tank.

The water heater simulation keys on two primary "states" of the water heater:

- Tank State – The tank can be in one of three states:
  - **FULL** – All the water in the tank is at a uniform temperature near the heater's setpoint. The One-Node model applies.
  - **PARTIAL** – The tank is in a state of partial depletion, where some of the hot water has been (or is being) drawn out, leaving hot and cold layers of water in the tank. The Two-Node model applies.
  - **EMPTY** – The tank has been completely depleted; all the water is at a uniform temperature near the water inlet temperature. The One-Node model applies.
- Load State – This refers to the current water load on the heater; that is, whether and how fast hot water is being drawn from the top of the tank. Formally, this load state applies only when the tank

state is **PARTIAL**, but is useful for the **FULL** and **EMPTY** tank states because it tells whether the tank will begin to move toward a **PARTIAL** state or stay in the current state. There are three possible load states:

- **DEPLETING** – Hot water is being drawn at a rate sufficient to move the boundary between the hot and cold zones upward. That is, hot water is being drawn out faster than the heating element can warm the incoming cold water, so the upper layer of hot water is getting smaller and the lower layer of cold water is getting larger.
- **RECOVERING** – Hot water is either not being drawn from the tank or begin drawn at a low enough level that the boundary between the hot and cold layers is moving downward. That is, the hot water is being drawn out at a rate low enough that the heating element can warm the replacement water faster than it is being introduced, causing the upper layer of hot water to get larger and the lower layer of cold water to get smaller.
- **STABLE** – Simplistically, this state implies that hot water is being drawn at a rate that matches the heating element's ability to warm the incoming cold water. Actually, the hot water draw does not have to exactly match the warming rate for this state to apply. Because there are other heat flows acting on the water (e.g., jacket losses to surrounding air), in any given situation where the tank state is **PARTIAL** there is a range of hot water flow rates for which the tank will never reach either the **FULL** or the **EMPTY** state.

For example, if the tank begins near the **FULL** state and hot water is drawn from it at a rate slightly higher than the heating element can match, the hot/cold boundary will begin to move upward. However, as the boundary moves up, the size of the hot layer gets smaller, reducing the area of warm tank that is exposed to the surrounding air. The lowered area means lowered heat losses, which may tip the balance so that the heating element eventually can keep up with the hot water draw. Similarly, starting with an almost empty tank, the heating element's capacity may exceed a small hot water draw and move the hot/cold boundary downward until jacket losses tip the balance. When the load state is **STABLE**, the calculated time to transition is infinite.

This **STABLE** state is illustrated in Figure 1. For an example water heater, the figure shows the time required to deplete an initially full tank ("time to transition") at various hot water flow rates. Note that flow rates above roughly 0.45 gpm result in positive and finite times to transition. Flow rates below about 0.44 gpm are finite and negative, meaning the tank is not depleting at all, but is actually recovering (i.e., the heating element can more than keep up with the heat removed by the water draw). In between those two points—between the positive and negative spikes to infinity—the tank is in the **STABLE** state that will never reach either the **FULL** or the **EMPTY** state.

These two critical flow rates depend on many factors, including the tank size and shape, tank UA, heating element capacity, and the hot and cold layer temperatures. They are identified in the GridLAB-D water heater model by calculating the rate of change of the hot/cold boundary position h with respect to time (dh/dt). Note in the lower graphic of Figure 1 that the time to transition is positive when dh/dt, calculated at the **FULL** starting point $h0$, is negative (meaning the tank is depleting). Also note that the time to transition is negative (meaning the tank is really recovering and will remain **FULL**) when dh/dt, calculated at the target **EMPTY** state, is positive. The area between the two critical flow rates is identified by differing signs between dh/dt calculated at **FULL** and dh/dt calculated at **EMPTY**.

Modeling Approach

Figure 2 shows a schematic representation of the water heater model in which Tavg is the average water temperature throughout the tank and Tamb is the ambient temperature. The thermal capacitance of the water Cw is a function of the tank volume:

$$C_w = V(gal)\frac{1(ft^3)}{7.48(gal)}\frac{62.4(lb_m)}{1(ft^3)}\frac{1(Btu)}{1(lb_m\dot{F}}$$

The thermal conductance of the tank shell (or "jacket") UA is calculated from the known R-values of the sides and top of the tank divided into their corresponding areas.

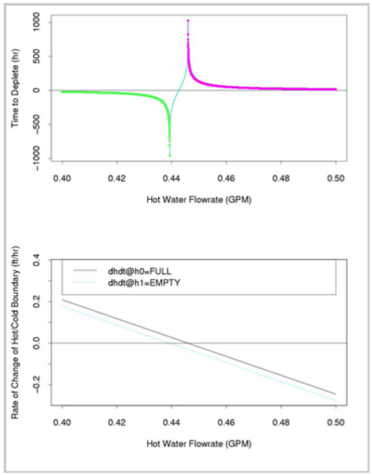

Figure 1 – Illustration of using dh/dt to identify the **STABLE** state

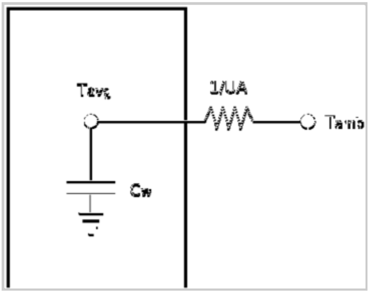

Figure 2: Water heater model schematic representation

One-Node Model

Considering Figure 2 and treating the water heater as a single node with thermal capacitance C_w, a conductance UA to ambient conditions, with mass flow rate and heat input rate of Q_{elec}, a heat balance on the water node is as follows:

$$Q_{elec} - \dot{m}C_p\left(T_w - T_{inlet}\right) + UA\left(T_{amb} - T_w\right) = C_w\frac{dT_w}{dt}$$

or,

$$dt = \frac{C_w}{\dot{m}C_pT_{inlet} + UAT_{amb} - \left(UA + \dot{m}C_p\right)T_w + Q_{elec}}dT_w$$

The time required to change the tank's temperature from an initial temperature T_0 to a new temperature T_1 is given by integrating that equation.

$$t_1 - t_0 = \int_{T_0}^{T_1}\frac{1}{\frac{\dot{m}C_pT_{inlet}+UAT_{amb}+Q_{elec}}{C_w} - \frac{UA+\dot{m}C_p}{C_w}T_w}dT_w$$

That is an integral of the form $dx/(a + bx)$, which has solution $\log(a + bx)/b$. Therefore, the final model of the time required to raise (or lower) the tank's temperature is

$$t_1 - t_0 = \frac{1}{b}\log(a + bT_w)\Big|_{T_0}^{T_1}$$

Where

- $a = \frac{\dot{m}C_p T_{inlet} + UAT_{amb} + Q_{elec}}{C_w}$
- $b = \frac{UA + \dot{m}C_p}{C_w}$

The reverse problem of calculating the new temperature of the tank from a known initial temperature and time difference, $t1 - t0$, follows directly:

$$T_1 = -\frac{a}{b} + \left(\frac{a}{b} + T_0\right) e^{b(t_1 - t_0)}$$

Two-Node temperature model

TODO: The two-node equations listed are incorrect, even though the repository code is correct. The latter should be parsed for the former. --Mhauer 20:11, 5 February 2009 (UTC)

This model, which applies when the heater is in a state of partial depletion, considers the heater to consist of two slugs of water, each at a uniform temperature. The upper "hot" node is near the heater's setpoint temperature, while the lower "cold" node is near the inlet water temperature. The time required to change the tank's hot water column from an initial height of h_0 to a final height of h_1 is given by the following equation:

$$t_1 - t_0 = \frac{1}{b}\log\left(\frac{dh_w}{dt}\right)\Bigg|_{h_0}^{h_1}$$

Where dh/dt is the location of temperature boundary along the height of the water column. This is calculated as a function of mass flow rate and the temperature difference across between the upper and lower interface layers of the water column, as given by the following:

$$\frac{dh}{dt} = a + bh$$

where

- $a = \frac{Q_{elec} + UAT_{amb}}{C_w T_l ower} - \frac{\dot{m}C_p}{C_w}$
- $b = \frac{UA}{C_w}$

In the two-node model, during each synchronization cycle, the height the hot water column is calculated based on the mass flow rate, using the following equation:

$$h_1 = \frac{e^{bT_w}(a + bh_0) - a}{b}$$

Simulation Sequence

TODO: Move this section to Dev:Residential --Dchassin 20:22, 24 November 2011 (UTC)

Each time the water heater is sync'd, the simulation follows four steps:

1. Calculate the energy consumed since the last iteration. The heater remembers whether it was heating and simply computes the consumption based on the time interval since the last sync.
2. Update the tank temperature or the location of the hot/cold boundary, depending on whether the tank was previously **FULL**, **PARTIAL**, or **EMPTY**.
3. Discern whether the tank needs heat. If the tank is in (or has reached) a **FULL** state at the thermostat setting, the power will be turned off. Otherwise the element will be turned on. Note that, for the Single-Node model, the heater state remains unchanged from its previous state when the water temperature is between the heating cut-off and cut-on temperatures (the deadband around the thermostat setpoint).
4. Calculate and post the time to next transition. For example, if the heater is on, this is either the time for the water to reach the cut-off temperature or for the hot/cold boundary to reach the bottom of the tank, depending on the tank state.

Complicating Factors

Several factors complicate the simulation. First, switching models between **FULL**, **PARTIAL**, and **EMPTY** states requires careful attention to not only what's happening now, but what state(s) things were in previously. Second, identifying when the load state is **STABLE** can be difficult. Finally, each of the models (One-Node and Two-Node) has two incarnations—one to calculate time to transition, and a second "inverted" one to calculate temperature/boundary location after a given amount of time. Because of limitations on floating point precision, the inverted model may not show that the water heater has actually reached the state it was anticipating. For example, the time model may show that the tank will reach the cut-off temperature of, say, 135 degreees in 12 minutes. After 12 minutes have elapsed, the core updates the water heater as expected. The temperature model may calculate that the final temperature is in fact 134.95 degrees. In a worst case the simulation could go into an infinite loop and never actually change states.

Lights

Residential lighting tends to be a mixture of several different types of lighting that is distributed somewhat randomly throughout the house and across the two circuit phases. Very little residential lighting is controlled by timers, although some outdoor lighting may be controlled by daylight sensors.

Depending on its location, the heat produced by a light fixture may go entirely into the conditioned space or may be partly dumped to an unconditioned space such as an attic or directly outdoors.

Modeling Assumptions

power_factor

If the *power_factor* is not set by the user, then the power factor depends on the lighting type:

- Incandescent: 1.00
- Fluorescent: 0.95
- CFL: 0.92
- SSL: 0.90
- HID: 0.97

voltage_factor

If the lights are not connected to a house circuit from which voltage can be determined, then the *voltage_factor* is 1.0.

shape.type

Only **analog** load shapes are supported. Any of three load shape scales are permitted

- Absolute (neither *shape.power* nor *shape.energy* are specified) and the schedule value is used as is
- Fixed energy (*shape.energy* is given) and the total energy used by any schedule block is made to made the *shape.energy* value.
- Scaled power (*shape.power* is given or *installed_capacity* is set) and the schedule value is multiplied by the *power*. This value must be provided if no schedule is given, and if missing, the *power_density* is used to compute the *installed_capacity* based on the *floor_area*. If the *floor_area* is not available, a default 2500 sf is used.

power_density

The default installed power density of each lights object is based on an assumed power density randomly chosen between 0.75 and 1.25 Watts/sf using a triangle distribution.

Note

The default installed power density approximates the lighting in a whole house, so is unlikely to be correct unless there is only one lights object in the model. Because most homes have a mixture of lighting types, and thus need more than one lights object, the default density may need to change in the future to be type-specific (which would imply that several lights objects would have to be specified to get a correct set of defaults).

heatgain_fraction

The fraction of lighting consumption that ends up as heat in the house defaults to 1.0.

curtailment

The fractional curtailment of the lighting (if any). The default value is 1.0 (i.e., no curtailment).

Modeling Approach

Each lights object will have a single type (incandescent, fluorescent, CFL, SSL, or HID) but may represent multiple fixtures in the house. It is expected that most models will contain one lights object for each type of lighting present in the house, although nothing constrains that.

Each object will have an installed capacity that represents its total consumption when all fixtures represented by the object are on. The scheduling of the lights object (on/off) is based on a simple demand multiplier (p.u.) that is read in from a tape . Future enhancements may give the lights module the ability to generate demand schedules in an intelligent (yet random) way.

Each object will have a constant power factor and each object will have a constant fraction of heat that enters the conditioned space. Thus, the real energy consumption is calculated as

$$P[kVA] = installed\_power[W] \times shape.load[pu] \times curtailment[pu] \frac{1[kw]}{1000[W]}$$

$$Q[kVAR] = P[kVA] \sqrt{\frac{1}{power\_factor^2} - 1}$$

Dishwasher

Dishwashers are fairly complex to control. Many wash cycles are simply timed prescriptions for filling, washing, releasing soaps and softeners, and drying, but most washers have several cycles to choose from. Some cycles in some dishwashers include a water temperature booster, which can be fairly energy intensive.

Modeling Assumptions

- Energy consumption in dishwashers is split between motors and resistance heaters. Thus the power factor changes depending on whether the wash cycle includes water temperature boost and changes from one part of the cycle to another. At this point, however, power factor modeling is simplified to be a constant value, fixed by default at 0.95.
- The fraction of washer consumption that becomes heat in the space is arbitrarily estimated to be 50%. This presumes that half the heat generated by the motors and/or resistance heater is transferred to the water and flushed down the drain.
- Installed capacity (installed_power) is randomly selected between 1000W and 3000W .

Modeling Approach

Dishwashers are modeled as a simple stream of energy demand signals taken from a tape. That is,

$$power(kW) = installed\_power \times demand/1000$$

The heat gains to the space are simply

$$internal\_heat(kW) = power \times heat\_fraction$$

Range

Electric ranges are essentially perfectly resistive loads. Each “burner” on a range top may be controlled by cycling the power on and off or by a “triac” device that regulates the on time at very high frequency. The user may reset the knob setting multiple times during a cooking event. Thus, range loads vary with the number of active burners, the user-controlled knob settings, and possibly the cycling behavior.

The GridLAB-D range model greatly simplified. It works in a manner essentially identical to the way plug loads are modeled—a demand (0 to 1) is read from a tape and multiplied by a range object-specific capacity (installed_power).

Modeling Assumptions

- Power factor is assumed to be fixed at 0.95.
- 100% of the range’s energy consumption is assumed to enter the house as heat (ignoring the heat that goes into the food).
- Installed capacity (installed_power) is randomly selected between 2500 W and 4500 W.

Modeling Approach

The range model retrieves its demand from a tape as a simple stream of values between 0 and 1 (representing the fraction of time the power is on). The demand is adjusted (scaled) by an “installed_power” value associated with each microwave (W). That is,

$$power(kW) = installed\_power \times demand/1000$$

The heat gains to the space are simply

$$internal\_heat(kW) = power \times heat\_fraction$$

Microwave

In their simplest mode of operation (full power), microwave ovens consume an essentially constant power. Other modes, such as defrost or partial-power, typically consume the same power in cycled on-off bursts. Microwaves are typically rated at 750 to 1100 Watts and operate at an efficiency in the neighborhood of 65%, meaning that 65% of the energy consumed goes toward heating food and the rest is released as heat to the surroundings. However the heat released to the surroundings is complicated. Although roughly 35% of consumed energy goes immediately to the house as heat, the 65% that warms the food is arguably released eventually as heat, either as the food cools on the table or as it is released as body heat from the humans who ate it.

Modeling Assumptions

- Power factor is assumed to be fixed at 0.95.
- Because microwaves have efficiencies in the range of 65%, approximately 35% of their

consumption goes immediately to the space as heat. Because body heat from occupants is theoretically accounted for separately, the microwave model assumes none of the remaining energy consumption becomes heat in the space .
- Installed capacity (*installed_power*) is randomly selected between 700 W and 2000 W .

Modeling Approach

The microwave model retrieves its demand from a tape as a simple stream of values between 0 and 1 (representing the fraction of time the power is on). Ordinarily, the values will be either 0 or 1 since most microwaves control cooking by cycling. The demand is adjusted (scaled) by an “installed_power” value associated with each microwave (W). That is,

$$power(kW) = installed\_power \times demand/1000$$

The heat gains to the space are simply

$$internal\_heat(kW) = power \times heat_f raction$$

Refrigerator

TODO: There are serious unit analysis problems in this section --Dchassin 00:22, 30 January 2009 (UTC)

The thermal and the electric loads on a refrigerator can be estimated using simplified first principles models. In this approach, the thermal load (primarily heat gain from ambient) is modeled as function of few lumped parameters (effective shell conductance, effective thermal mass, and compressor efficiency), ambient condition, internal gains adding and removing food material, and thermostat setting. Once the thermal load is estimated the power consumption of the compressor and the fan can be calculated. Although this approach is not as detailed and accurate as the detailed physical model, it does provide reasonably accurate estimates of the energy consumption.

Load Calculation

The major heat gains that contribute to the refrigerator thermal load are:

1. Conduction through refrigerator/freezer walls.
2. Heat gain from infiltration of ambient air when the refrigerator/freezer door is opened.
3. Heat gains from additions of food to the refrigerator/freezer.

In general, the amount of heat that must be removed (cooling load) is not always equal to the amount of heat received at a given time. The difference is a result of the heat storage and time lag effects. Only a portion of the heat entering refrigerator actually cools the air inside the refrigerator immediately; the rest cools the mass – the food material. The heat that is stored in the mass will result in thermal load at a later time. So, the modeling approach will have to account for the storage effect.

Modeling Assumptions

- Power factor is assumed to be fixed at 0.95.
- The thermal conductance of refrigerator and freezer compartments are assumed to be random normal in the range of 0.9 to 1.1 Btu/h.f 2.°F
- The refrigerator compartment set point is assumed to be between 35 and 39°F.

Modeling Approach

The refrigerator model is currently implemented in its simplest form to calculate the time to change of state based on the current indoor temperature, refrigerator interior temperature, thermal properties of the casing, and food content and water content of the refrigerator compartment. The original ODE relating these properties is

$$\frac{C_f}{UA_r + UA_f} = \frac{dT_{air}}{dt} + T_{air} = T_{out} + \frac{Q_r}{UA_r}$$

where

- T_{air} is the temperature of the air
- T_{out} is the ambient airtemperature around the refrigerator
- UA_r is the UA of the refrigerator itself
- UA_f is the UA of the food-air
- C_f is the heat capacity of the food
- Q_r is the heat rate from the cooling system

The general solution is:

$$T_t = (T_0 - C_2)\, e^{-\frac{t}{C_1}} + C_2$$

where

- t is the elapsed time
- T_0 is the initial temperature
- T_t is the temperature at time t
- $C_1 = C_f/(UA_r + UA_f)$
- $C_2 = T_{out} + Q_r/UA_f$

The time solution is:

$$t = -ln\frac{T_t - C_2}{T_0 - C_2}C_1$$

During each synchronization cycle, the refrigerator model calculates the internal gain based on the rated capacity of the refrigerator and returns the time solution for determining the next synchronization time.

Internal Gains

Each house modeled in GridLAB-D has two primary sources of internally generated heat. First, is the "waste" heat given off by other devices in the house that are modeled by GridLAB-D. For example, a refrigerator located inside the house is modeled primarily for its thermal behavior regarding cooling and freezing food, with its primary output being the impact (energy and peak) on the grid. However, the refrigerator also gives off heat that must be accounted for in the simulation of the house itself. Second, there are devices in the home that are not explicitly modeled by GridLAB-D, but that have sufficient aggregate impact that their heat contribution to the house and their aggregate impact on the home's electricity must be accounted for. Examples are TV sets, vacuum cleaners, hand-held hair dryers, and other miscellaneous equipment generally plugged into wall outlets.

Modeling Assumptions

- All "plug loads" are assumed to be consolidated into a single load. Thus, they are all either on one side of the home's circuit split or evenly distributed between the two sides; this is determined randomly at object creation.
- The installed capacity of plug loads is randomly selected at a value between 700 and 2000 Watts.
- The fraction of plug load consumption that ends up as heat in the house is fixed at 90%.
- The power factor of the aggregate plug loads is fixed at 0.95.

Modeling Approach

The internal gains reported by modeled devices are simply collected and summed at the house level, the house object being unaware of any details other than the consumption, circuit, and power factor reported by each device.

Other internal gains (aka plug loads) are modeled as a simple, fixed capacity (installed_power) that is multiplied at each time step by a demand fraction (p.u.) read from a tape. [**TODO:** Is this right? The code currently sets the demand as a random value between 0 and 0.1. --Dchassin 00:26, 30 January 2009 (UTC)] That is,

$$power(kW) = installed\_power(W) \times demand/1000$$

Further, the heat to the surrounding space is given by

$$internal\_heat = power \times heat\_fraction$$

See Also

- Modules
- Residential
  - House
  - Appliances
    - Refrigerator
    - Range
    - Dishwasher

- Clotheswasher
- Dryer
- Microwave
- Freezer
- EV charger
- Lights
- Plugs

References

Pita, G. Eward. 2002. Air-Conditioning Principles and Systems. Prentice Hall, Upper Saddle River, New Jersey, 07458.

Pratt, R.G. and Z.T. Taylor, 1994. "Development and Testing of an Equivalent Thermal Parameter Model of Commercial Buildings from Time-Series End-Use Data," Pacific Northwest Laboratory, Richland, Washington.

Pratt, R.G., et al., 1990. "Significant ELCAP Analysis Results: Summary Report," PNL-6659, Pacific Northwest National Laboratory, Richland, Washington.

Sonderegger, R., 1978, "Dynamic Models of House Heating Based on Equivalent Thermal Parameters," Report PU/CES 57, Doctoral dissertation, Princeton University, Princeton, New Jersey.

Subbarao, K., 1981, "Thermal Parameters for Single and Multizone Buildings and Their Determination from Performance Data," Solar Energy Research Institute, Golden, Colorado.

Wilson, N.W., B.S. Wagner and W.G. Colborne, 1985, "Equivalent Thermal Parameters for an Occupied Gas-Heated House," ASHRAE Transactions, vol. 91, part 2.

See also

- Residential module
  - User's Guide
  - Appliances
  - house class – Single-family home model.
  - residential_enduse class – Abstract residential end-use class.
  - occupantload – Residential occupants (sensible and latent heat).
  - ZIPload – Generic constant impedance/current/power end-use load.
- Technical Documents
  - Requirements
  - Specifications
  - Developer notes
  - **Technical support document**
  - Validation