Commercial Load Module

user documentation

Prepared by: Battelle Columbus Operations

505 King Ave.

Columbus, OH 43202

Date: 10 October 2014

Abstract

This document provides documentation of the commercial load module developed for use with GridLAB-d.

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Organization/Point of Contact | Description of Changes |
| 1.0 | 10/10/2014 | Anthony Kuhlman | Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[REVISION HISTORY 2](#_Toc400960589)

[Version 2](#_Toc400960590)

[Date 2](#_Toc400960591)

[Organization/Point of Contact 2](#_Toc400960592)

[Description of Changes 2](#_Toc400960593)

[1. OVERVIEW 4](#_Toc400960594)

[Purpose 4](#_Toc400960595)

[2. Overview of operation 5](#_Toc400960596)

[3. Using the commercial load object 7](#_Toc400960597)

[Appendix A. Sample model using the commercial load object 11](#_Toc400960598)

# 1. OVERVIEW

The Commercial Load Module was developed by Battelle Memorial Institute in collaboration with Southern California Edison (SCE) for use in the GridLAB-D power distribution system simulation tool. The aim of the commercial load module is to provide realistic simulation of commercial customers

## Purpose

This document provides an introduction to the commercial load module, including a high level discussion of how the module estimates the load for a given customer. This document also will also illustrate the use of the commercial load object within a GridLAB-D model.

A detailed discussion of the analysis performed to arrive at the underlying model used by the commercial load object is provided in a separate document.

# 2. Overview of operation

The underlying behavior of the Commercial Load Module is governed by a regression equation that considers the customer’s building type, climate zone, North American Industry Classification System (NAICS) code, and average daily energy usage, as well as the current date, time, and outdoor temperature to compute a load value. The building type, climate zone, and NAICS code are used to select a set of coefficients that describe the time- and temperature-dependent behavior of the customer’s load. The average daily usage serves to scale the calculated load value.

At startup, the commercial module is initialized by loading in the regression equation coefficients from the supplied comma-delimited coefficient file. Then, each commercial\_load object selects an appropriate set of coefficients from these values based on the specified building type, climate zone, and NAICS code.

At each time step, each commercial\_load object uses the regression equation to compute a load value for the customer. This value is assumed to be the total power consumed by the customer during that time step. This load is divided equally over the connected phases of the object. For example, if the object is connected to all phases (ABC), then the calculated value is divided by 3 and 1/3 of the total load is put onto each phase.

In much the same way as the powerflow::load object originally supplied by GridLAB-D, the commercial load object allows the user to divide the calculated load into current, impedance, and power loads by specifying the fraction of the load to use for each type. These fractions can be specified for each phase. Additionally, for each phase a power factor may be specified for each type of load to allow users to further refine the behavior of the load object.

In these terms, the apparent power for each phase is given in terms of the total apparent power by

For each phase, the constant power load is given by

where the power fraction of the load for phase i is specified by and the power factor for this portion of the load is given by .

The constant current and constant impedance portions of the load are computed similarly, i.e.,

and

where is taken to be the nominal voltage of the load. We have assumed that the nominal voltage is real valued.

# 3. Using the commercial load object

The first step in using the commercial load object in a GridLAB-D model is to ensure that the commercial module is loaded by GridLAB-D at the start of the simulation. This is accomplished by adding the block

to the module definition section of the glm file. “<path to csv file>” should be replaced with the path to a comma delimited file that contains the regression coefficients for the underlying model (as discussed above). This file path must be specified for the module to initialize correctly. Table 1 lists the parameters that may be specified in the module definition block.

Table . Module parameters for the commercial module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter name | Units | Required | Default | Description |
| coefficient\_file | *unitless* | Yes |  | Path to a comma delimited file containing the regression coefficients for the commercial load model. |

A commercial load is created by specifying a commercial\_load object. The available parameters for this object type are listed in Table 2. For a working model, it is necessary to specify the parent, average daily energy usage, and the phases of the object. In this case, the remainder of the parameters will be set to their default values. To more accurately simulate a specific load, it is necessary to supply the building type and climate zone. The highest degree of accuracy is obtained by supplying building type, climate zone, and NAICS code.

Table . Class parameters for commercial\_load objects.

| Parameter name | Units | Required | Default | Description |
| --- | --- | --- | --- | --- |
| name | *unitless* | No |  | Name of object |
| parent | *unitless* | Yes |  | Parent object of this load. This object should be of type powerflow::node (node, meter, triplex\_meter, etc.) |
| building\_type | *unitless* | No | 1 | Building type for this load. These types are taken from SCE data. To agree with the current coefficient file, this value should be in the range [1, 33], excluding 5, 16, 22, 24, and 30. |
| climate zone | *unitless* | No | 6 | Climate zone for this load. This is the SCE defined climate zone for the customer and can be one of (6, 8, 9, 10, 13, 14, 15, 16). |
| naics\_code | *unitless* | No | -1 | NAICS code for this load. If this value is not found in the supplied coefficients file, the object will use the default set of parameters for the given building type and climate zone. |
| power\_fraction\_A | *unitless* | No | 1 | Fraction of the power on phase A that will be assigned to a constant power load. It is expected that , though this is not currently enforced. |
| power\_fraction\_B | *unitless* | No | 1 | Fraction of the power on phase B that will be assigned to a constant power load. |
| power\_fraction\_C | *unitless* | No | 1 | Fraction of the power on phase C that will be assigned to a constant power load. |
| current\_fraction\_A | *unitless* | No | 0 | Fraction of the power on phase A that will be assigned to a constant current load. |
| current\_fraction\_B | *unitless* | No | 0 | Fraction of the power on phase B that will be assigned to a constant current load. |
| current\_fraction\_C | *unitless* | No | 0 | Fraction of the power on phase C that will be assigned to a constant current load. |
| impedence\_fraction\_A | *unitless* | No | 0 | Fraction of the power on phase A that will be assigned to a constant impedance load. |
| impedence\_fraction\_B | *unitless* | No | 0 | Fraction of the power on phase B that will be assigned to a constant impedance load. |
| impedence\_fraction\_C | *unitless* | No | 0 | Fraction of the power on phase C that will be assigned to a constant impedance load. |
| power\_pf\_A | *unitless* | No | 1 | Power factor for constant power load on phase A. |
| power\_pf\_B | *unitless* | No | 1 | Power factor for constant power load on phase B. |
| power\_pf\_C | *unitless* | No | 1 | Power factor for constant power load on phase C. |
| current\_pf\_A | *unitless* | No | 1 | Power factor for constant current load on phase A. |
| current\_pf\_B | *unitless* | No | 1 | Power factor for constant current load on phase B. |
| current\_pf\_C | *unitless* | No | 1 | Power factor for constant current load on phase C. |
| impedence\_pf\_A | *unitless* | No | 1 | Power factor for constant impedance load on phase A. |
| impedence\_pf\_B | *unitless* | No | 1 | Power factor for constant impedance load on phase B. |
| impedence\_pf\_C | *unitless* | No | 1 | Power factor for constant impedance load on phase C. |
| nominal\_voltage | V | No | 240V | Nominal voltage for the load. |
| average\_daily\_usage | kWh | Yes |  | Average daily energy usage for this load. This value has the effect of scaling the overall load. |
| phases | *unitless* | Yes |  | Phases used by this load. This value can be any combination of A, B, and C. |

Note that the parent of a commercial\_load object is required to be an object of type powerflow::node or a type that derives from node. Recommended types for the parent object are node, meter, and triplex\_meter. Other types that derive from node (e.g., load, motor, or capacitor) *should* work as well, but this has not been tested.

To obtain temperature dependent behavior, it is necessary to include a climate object in the model that will provide the outdoor temperature at a given time step. If no climate object is found, the commercial module will assume a constant outdoor temperature of 18° C. This effectively removes the temperature dependence of the commercial load module.

A complete example that uses the commercial\_load object is given in Appendix A.

# Appendix A. Sample model using the commercial load object

|  |
| --- |
| #set minimum\_timestep=60;  clock {  timezone EST+5EDT;  starttime '2010-01-01 0:00:00';  stoptime '2010-01-04 0:00:00';  }    module climate;  module tape;  module powerflow {  NR\_iteration\_limit 1000;  solver\_method NR;  }  module commercial {  coefficient\_file coefficients.csv;  }  object commercial\_load{  name commload;  parent tm1;  building\_type 8;  climate\_zone 6;  naics\_code 445120;  power\_pf\_A 1.0;  power\_pf\_B 1.0;  power\_pf\_C 1.0;  power\_fraction\_A 1.0;  power\_fraction\_B 1.0;  power\_fraction\_C 1.0;  average\_daily\_usage 81.212;  phases ABC;  nominal\_voltage 480.0;  }  object recorder {  name load-recorder;  file comm-load-recorder.csv;  interval 900;  parent commload;  property power\_A, power\_B, power\_C;  }  object node {  name transmission\_node;  phases ABC;  bustype SWING;  nominal\_voltage 4160;  }  object climate {  name "Columbus OH";  interpolate QUADRATIC;  tmyfile "Columbus\_TMY2.tmy2";  }  object transformer\_configuration {  name tc400;  connect\_type WYE\_WYE;  install\_type PADMOUNT;  power\_rating 500;  powerA\_rating 167;  powerB\_rating 167;  powerC\_rating 167;  primary\_voltage 4160;  secondary\_voltage 480;  resistance 0.011;  reactance 0.02;  }  object recorder {  name xfmr\_recorder;  file xfmr\_recorder.csv;  interval 900;  parent transformer1;  property power\_out\_A, power\_out\_B, power\_out\_C;  }  object transformer {  phases ABCN;  name transformer1;  from transmission\_node;  to n1;  configuration tc400;  }  object node {  name n1;  nominal\_voltage 4160;  phases ABCN;  }  object recorder {  name node\_recorder;  file node\_recorder.csv;  interval 900;  parent n1;  property voltage\_A, voltage\_B, voltage\_B;  }  object underground\_line\_conductor {  name ulc6060;  outer\_diameter 1.290000;  conductor\_gmr 0.017100;  conductor\_diameter 0.567000;  conductor\_resistance 0.410000;  neutral\_gmr 0.0020800;  neutral\_resistance 14.87200;  neutral\_diameter 0.0640837;  neutral\_strands 13.000000;  insulation\_relative\_permitivitty 2.3;  shield\_gmr 0.000000;  shield\_resistance 0.000000;  }  object line\_spacing {  name ls515;  distance\_AB 0.500000;  distance\_BC 0.500000;  distance\_AC 1.000000;  }  object line\_configuration {  name lc606;  conductor\_A ulc6060;  conductor\_B ulc6060;  conductor\_C ulc6060;  spacing ls515;  }  object underground\_line { //684652 {  phases ABC;  name ugline;  from n1;  to tm1;  length 800;  configuration lc606;  }  object meter {  name tm1;  nominal\_voltage 4160.0;  phases ABC;  }  object recorder {  name meter\_recorder;  file meter\_recorder.csv;  interval 900;  parent tm1;  property measured\_power\_A, measured\_power\_B, measured\_power\_C;  } |