**12º CONGRESSO INTERNACIONAL DE
AR CONDICIONADO, REFRIGERAÇÃO, AQUECIMENTO E VENTILAÇÃO**

RUNNING A LARGE NUMBER OF ENERGY SIMULATIONS USING THE BATCH PROCESSING FEATURE OF DOE-2 EQUEST

Pedro Guaraldi - pedroguaraldi@gmail.com
Concordia University, Montréal, Canada, www.concordia.ca
Radu Zmeureanu - radu.zmeureanu@concordia.ca
Concordia University, Montréal, Canada, www.concordia.ca

F3 - Cálculo, Projeto e Simulação Termo Energética

Abstract. *This paper proposes a method of running a large number of building energy simulations using the batch processing feature of the eQUEST program. Users can benefit from executing limitless consecutive simulations, starting from a database that contains instructions to be passed to the DOE-2.2 engine. The instructions are used to modify specific parameters, generating new models from a building energy model base (reference) file. The process of creating the instructions database is described in detail. The method is demonstrated in a case study institutional building in Montréal, Canada. The impact of oversizing HVAC equipment and systems is evaluated through whole-building and system level energy performance indicators. By using this feature, the effort otherwise required to analyse the impact of many permutations of design decisions on the building energy performance is considerably reduced. In addition, the results available in the automatically generated output file are useful to detect trends and allow for a rapid evaluation of the simulation runs.*

Keywords: *Building energy simulation, consecutive simulations, batch processing, eQUEST, oversizing of HVAC equipment*

1. INTRODUCTION

Simulation programmes have been available since the 1960s, aiding in the design of buildings by providing load calculations and energy analysis. The drive for a more sustainable built environment has prompted the development and improvement of these programmes. Capabilities to model heat and mass transfer, airflow, daylighting and different types of system and components were successively appended. Improved graphical user interfaces have facilitated the use of complex features and have contributed to the success and widespread use of these tools (Clarke & Hensen, 2015).

Since its introduction and popularization in the building engineering field, these programs, commonly referred as Building Energy Modelling (BEM) or Building Performance Simulation (BPS) tools, have been used in a broad spectrum of applications. These programs facilitate the modelling of building composition and utilisation and provide engineers with useful and easy-to employ energy diagnostics tools. Hong, Chou and Bong (2000) listed some of the popular applications: (i) building heating/cooling load calculation; (ii) energy performance analysis for design and retrofitting; (iii) Building Energy Management and Control System (EMCS) design; (iv) compliance with building regulations and standards; (v) cost analysis; (vi) study of passive energy saving options, and (vii) Computational Fluid Dynamics.

Amongst the objectives of using BEM/BPS programmes, one can cite: (i) to ensure requisite levels of comfort and indoor air quality; (ii) to devise energy efficiency and demand management solutions; (iii) to embed new and renewable energy technologies; (iv) to lessen environmental impact; (v) to ensure conformance with legislative requirements, and (v) to formulate energy action plans at any scale. One approach used to accomplish these objectives is to use BEM/BPS programmes to evaluate design variants iteratively and identify new potential directions in the development process. These programs can also be used to evaluate the robustness of a technology under different usage scenarios and operating conditions. Lastly, they provide an opportunity to assess materials, components and systems in a virtual environment without the need to build physical test prototypes. These tools equip engineers with a cost-effective method to explore different building design options and technologies. With uses exceeding the design phase of buildings, BEM/BPS programmes can also assist in the phases of building commissioning and operation. During these phases, comparisons between predicted and measured performance and evaluation of whole life performance aggregate robustness to the analysis (Clarke & Hensen, 2015).

Substantial energy savings during building operation can be achieved when conventional building design considers energy efficiency. HVAC systems and lighting relate to a large portion of a building's energy use and are good candidates to be the focus of energy efficiency measures. BEM/BPS programmes can be used to model such systems individually and as parts of an integrated system, modelled to represent complex and dynamic interactions between the outdoor environment, the building envelope, the HVAC system, the lighting system, and control devices (Hong et al., 2000).

There are several BEM/BPS programs available on the market with diverse sets of features that render different scopes. Rallapalli (2010) reiterates that the choice of the simulation program is dependant on the usability and applicability of the program with respect to the user's needs. The author presents a comparison between eQUEST and EnergyPlus modelling process and simulation results for a medium sized office building. As main characteristics for eQUEST, a user-friendly graphical interface and rapid simulation times for a high number of spaces are pointed out. As for EnergyPlus, it is pointed out the capacity to model more complex systems, producing more accurate results, but with longer simulation times.

Tian et al. (2017) use the EnergyPlus program to carry out batches of building energy simulation. The authors point out that the EnergyPlus program was chosen since the IDF files (input data files) are ASCII files (unformatted text file) that can be easily edited using computer languages such as R, MATLAB and Excel VBA. Scripts created using these computer languages are used to automatically generate hundreds or thousands of energy models that are variations of a base case IDF file. Variations of parameters in input files and the results obtained from the simulation of the energy models are then analysed to assert which parameters have the highest impact in selected Key Performance Indicators (KPIs).

In simulation applications, it is often necessary to perform sequential simulations to evaluate different scenarios. These scenarios encompass different design characteristics, such as a range of building envelope thermal characteristics, building types and climatic regions. Regarding the process of creating building models and collecting the results, Tian (Tian, 2013) points out the need of an automated process due to the analysis requiring many simulation runs to be performed. Clarke and Hensen (2015) highlight that the computational burden of such studies rises with the need to perform multiple sequential simulations. Building simulation programs which provide flexibility, easily processed input files and satisfactory simulation runtimes, such as EnergyPlus, ESP-r, TRNSYS, DOE-2 and eQUEST can be used as tools to facilitate the creation of these models.

This paper presents a method of running a large number of building energy simulations using the batch processing feature of the widely used building energy simulation program developed by the Unites States D.O.E. (Department of Energy), eQUEST. The simulation program and the proposed method were chosen since they are easily implemented and time efficient and use readily available computational resources. The method is applied in a case study institutional building, modelled to be compliant with NECB-2017 (Natural Resources Canada, 2017). The labor saving and straightforward approach advantages of the proposed method are demonstrated in a case study evaluation of the impact of oversizing HVAC equipment and systems using whole-building and system level energy performance indicators. This evaluation is performed with the aid of the automatically generated output file of the batch processing feature.

2. METHOD

The proposed method of batch simulations and analysis of sensitivity is comprised of the following steps: (1) Creation of a case study (reference) building energy model using the eQUEST program; (2) Selection of the design input variables, and assignment of numerical value ranges for each variable; (3) Selection of the KPIs; (4) Creation of a Batch Master table and the required slave tables containing the information regarding the input variables to be passed on to the eQUEST program (Madison, 2012); (5) Execution of the batch simulations; (6) Extraction and organization of the results; (7) Calculation of the derived KPIs; (8) Calculation of the sensitivity coefficients, and (9) Analysis of the results, discussion and conclusions.

2.1. Building Energy Model Development

The chosen building to perform this case study is the Centre for Structural and Functional Genomics building, a research facility of Concordia University. This research facility comprises mainly office and laboratory spaces served by ancillary conference rooms, storage rooms and corridor spaces. The building envelope, loads and systems are modelled to fully embody the energy efficiency requirements of NECB-2017. The fully compliant case study reference building model serves as a benchmark to evaluate the impact of oversizing HVAC equipment and systems on selected KPIs.

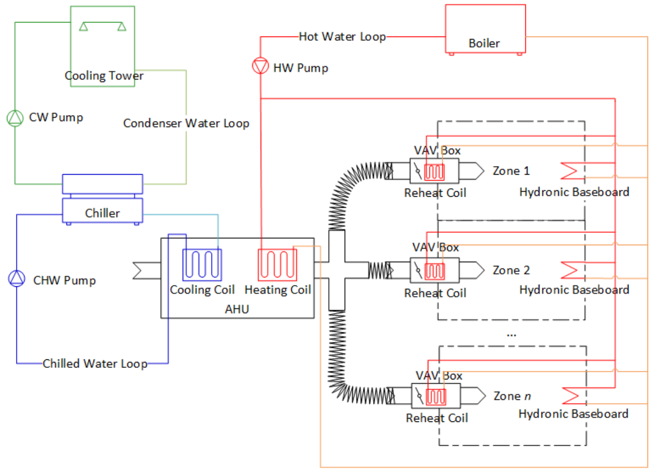

Figure 1. Water-side HVAC System representation.

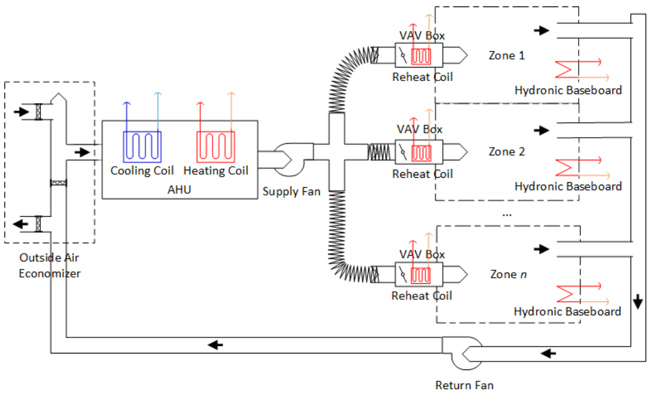

Figure 2. Air-side HVAC System representation.

A representation of the water-side system modelled in eQUEST is presented in Figure 1. The chilled water loop comprises a water-cooled water chiller, a chilled water (CHW) pump and a central cooling coil. The condenser water loop consists of a cooling tower and condenser water (CW) pump. The hot water loop consists of a natural gas-fired boiler, a hot water (HW) pump, a central heating coil and a set of baseboard heaters and reheat coils installed in each zone. A representation of the air-side system modelled in eQUEST is presented in Figure 2. The system is a multi-zone Variable Air Volume (VAV) system with VAV boxes with reheat. An outdoor air economizer is also present.

This model is simulated using the CWEC weather files for Montréal and includes both heating and cooling seasons (Numeral Logics, 1999).

2.2. Selection of Design Input Variables

The classes of design and operating variables used as inputs in the sensitivity analysis are listed in Table 1. The base case corresponds to prescriptive and performance requirements relating to the building envelope (Part 3 of NECB-2017), HVAC Systems (Part 5 of NECB-2017) and compliance to the whole-building energy performance (Part 8 of NECB-2017). Each one of the 15 variables are varied one at a time for 5 times, creating an input space of 75 building energy models. The 5 different values used to oversize the systems correspond to models with oversizing factors of 0%, 10%, 20%, 30% and 40%.

Table 1. Classes of input variables used in the simulations

Loops	Primary Equipment	Secondary Equipment	Pump Flow	Pump Head
Chilled Water Loop	Chiller	VAV System	Chilled Water Pump	Chilled Water Pump
Hot Water Loop	Boiler	VAVS Heating Coil	Hot Water Pump	Hot Water Pump
Condensed Water Loop	Cooling Tower	VAVS Cooling Coil	Condensed Water Pump	Condensed Water Pump

2.3. Selection of Key Performance Indicators

The KPIs are calculated from the simulation results at the whole-building and system level. The impact of oversizing of HVAC systems and equipment on the total annual energy consumption from all energy sources of the building is evaluated through the **Energy Use Intensity per floor area** (GJ_{en}/m^2) KPI. In addition, the impact on the energy performance of the building is evaluated through the **System Performance Ratio of both Heating and Cooling Systems** (kW_{th}/kW_{en}) KPI. This KPI is the coefficient of performance of the entire system, including main equipment (e.g. chillers, boilers) and ancillary equipment (e.g. cooling towers, fans, pumps). The subscript *en* refers to the energy input to an equipment, system, plant, space, zone or building, while the subscript *th* refers to the thermal energy output.

2.4. Batch Processing in eQUEST

The batch processing feature in eQUEST is a native functionality that allows users to execute limitless consecutive simulations. Tools that automatize the simulation process are of paramount importance in BES/BPS due to the need of exploring the relationship between inputs and outputs over a range of scenarios. This implies the availability of hundreds or thousands of simulated models representing different combinations of input parameters. This subsection develops on scarcely available documentation and provides users with the knowledge needed to increase their simulation productivity in eQUEST. The step-by-step process of set-up and utilization is explained with an accompanying illustration and example files.

This feature relies on an easy to build database that contains instructions to be passed to the DOE-2.2 engine to modify specific parameters, generating new models from a base case model file. By using this feature, the effort otherwise required to generate numerous simulation models representing different design scenarios is considerably reduced. After the models are generated and simulated, an automatically generated Comma Separated Value (.csv) output file containing some main building performance indicators and check figures is created. This automated model generation, simulation and result extraction feature allows users to analyze the impact of many permutations of design decisions on the building energy performance in a time and resource-efficient way.

One of the shortcomings of the batch processing feature is that up to the current version (eQUEST 3.65, build 7175), there is no official documentation available on how to set up and run the feature. Users rely on experimentation and expertise from other users to develop proficiency in using this tool (Madison, 2012). To address this shortcoming, the process of setting up eQUEST to perform batch processing is detailed in this section, with a general overview of the tool approach illustrated in Figure 3.

The process starts with a series of .csv tables ("Tables.csv"), called data modification tables, that contain specific instructions on what properties or parameters are to be modified to generate each new model, and to which values these properties and parameters should be modified to. These "slave" tables of data modifications are then passed on to a "master" batch directives table.

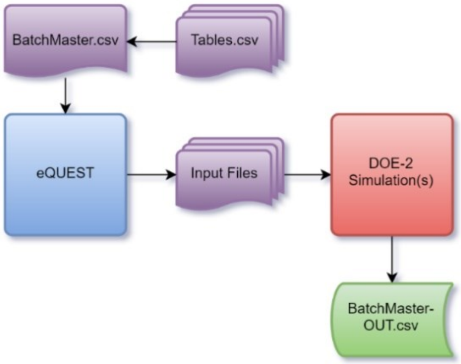

Figure 3. Batch Processing Work-flow (Lerond, 2017)

The master batch directives table ("BatchMaster.csv") contains global information on the batch simulation run. It indicates what type of processing is to be performed (simulation, ranked efficiency measure runs, etc.), what is the measure analysis information (building type, climate zone, etc.), which tables of data modifications are to be used, and it contains the file locations of the existing and new model simulations to be executed.

After these two tables are created, the location of the master table file is passed on to eQUEST, which then uses the instructions contained in that table to generate input files that will be used for the simulation of the new variation case models from the base case model. These new variation case models are then passed on as input files to the DOE-2.2 simulation engine. Once the simulations are performed, some main performance indicators and check figures are compiled into an output table file ("BatchMaster-OUT.csv").

This method can also be used to determine the sensitivity of the building performance with relation to its systems and components. To perform this task, a MATLAB code was implemented to extract and calculate additional KPIs from the simulations output files.

Using the Tool

Once a base case model file is opened in the eQUEST graphical interface, the feature can be accessed under "*Tools → Batch Processing…*". Once the feature is selected, a prompt is shown requesting the user to provide two inputs: the Batch Directives and the Batch Output Results file locations. The Batch Output Log file is saved in the same location as the Batch Output Results File. The accepted file format for the Batch directives file is the CSV format. For ease of use, these files can be created using a spreadsheet format with an appropriate program, such as Microsoft Excel, and saved with the .csv file extension. Once the file locations have been provided, the user can start the batch processing by clicking on the *"Run"* button.

The batch processing status screen is shown when the process is started. In this screen, the progress of the batch simulations can be monitored. In each new model run, a new set of DOE-2.2 input files (with file extensions .inp and .pd2) and all their supporting files are created and saved into a previously defined location in the master batch directives table. This new set of files is comprised of a modified version of the base case file according to the instructions in the master table file that reference the tables of data modification (e.g. with different geographical locations, wall thermal transmittances, equipment efficiencies, etc.). Then, the simulation itself for the newly created model is performed. All of these events are recorded in the status screen, preceded by a time stamp.

At the end of a successful batch simulation run, the status screen concludes the processing with a message stating the total number of records processed successfully. An output .csv file summarizing the results of the batch simulations is then automatically generated. Each row refers to a specific record, and each column refers to a property of said record. This standardized output file contains information concerning each simulation run, compiled from the DOE-2.2 reports. The detailed process used to create the batch directives file and the table of data modification files is described in the next sections.

Batch Directives File Creation Process

The process of creating the batch directives file is comprised of the following subprocesses: (1) Define Global Information; (2) Define Measure Analysis Information; (3) Define Tables of Data Modifications, and (4) Define Batch Processing Actions.

The first sub-process in the process flow of the batch directives file creation is the **Define Global Information**. In this sub-process the batch version and type properties are defined. The batch version is a number denoting which version of the batch processing mechanism the file is written in. The batch type is only specified for batch versions higher than "1" and determines what type of processing is to be performed. For versions higher than "1", it is possible to perform more advanced analyses.

Options for these more advanced analyses include performing one or more DOE-2 simulations including wizard-generated models and compliance analysis (*Batch Type = Simulation*), generating a Building Description Language (BDL) library of different components (*Batch Type* = *WindowLibrary*), and performing efficiency measure runs in which each baseline model is run with different measures to rank the impact of the measures on the energy performance of the building (*Batch Type* = *RankedEffMeasures*). If the user wishes to perform these analyses, additional information must be provided in the second sub-process, named **Define Measure Analysis Info**. Other batch types (*Batch Type* = *EfficiencyMeasures)* that were under development and had not been yet implemented at the time the supporting literature was published might be available.

For the case study, the focus of the analysis is to determine the sensitivity indices of the output variables related to the input variables. The baseline model was created beforehand, alongside with the data modification tables necessary for creating the variation cases for the sensitivity analysis. For this reason, the batch version "1" was chosen since it provides all the necessary tools for this study.

In the third sub-process **Define Tables of Data Modifications,** information concerning the instructions to generate the variant models is provided. The variant models are generated from a base case model file after performing changes in two categories: changes to the Project/Wizard/Energy Efficiency Measures component (stored in the *.pd2* file of the model) and changes to the BDL component (stored in the *.inp* file of the model).

The *.pd2* file contains all properties that regard either the building geometry or the project description. Examples of these properties are the weather file, the location of the building and geometrical positions of floors, walls and windows. The *.inp* file contains all the information regarding the building description and is written in Building Description Language (BDL). Examples of these properties are global parameters used in the simulation, materials, layers and constructions that are parts to the building shell, glass types and schedules.

In this sub-process, for each row of the batch directives file, the number of *.pd2* file data modification tables and the number of *.inp* file data modification tables are written, alongside with the names of these tables. With these parameters defined, the batch directives file is able to reference the other data modification tables and modify the base case model to generate the new variant model files.

The fourth and last sub-process is the **Define Batch Processing Actions**. In this sub-process, properties are defined for each desired combination of modifications. Each row in this part of the table contains information regarding a new record and represents a new variant building model to be generated from the base case model and simulated.

In each row a process record flag, determining if the record should or should not be processed is defined. Then, a '*new or existing*' flag is defined, with the option to either create and save a new set of project files or merely use the current one. For both options, the file location must be provided. If the user wishes to perform compliance analysis using a compliance ruleset file, the user can provide the name of the ruleset or use the '*unchanged'* property not to do compliance analysis. Following these steps, a number of columns is created, one for each property defined in the tables of data modification. For each column, a number is given that points to the row containing the value in the table of data modification the user wishes this property to be set to. Next, a wizard to BDL conversion flag is defined, determining whether inputs converted from the wizard data should be used based on if they were previously edited using the table of modifications data. The type of the analysis to be performed is defined and lastly, the format of the output file is defined.

Example of Batch Directives File

An example of a batch directives CSV file generated according to the instructions presented in the previous section is reproduced in the greyed text box below. This example is a shortened and simplified version of the file used in the case study.

```
1
0
3,tbl_ChillerOversize,tbl_BoilerOversize,tbl_
CTOversize
1,C:\SampleBaseProject,C:\SampleModifiedProje
ct-A,unchanged,2,4,3,0,Simulate,CumCSV
1,C:\SampleBaseProject,C:\SampleModifiedProje
ct-B,unchanged,1,3,5,0,Simulate,CumCSV
-1
```

Figure 4. Example of Batch Directives File

```
Batch Ver
Num Tbls, Proj/Wiz/EEM mod Table #1, ...
Num Tbls, BDL mod Table #1, BDL mod Table #2,
...
Proc Rec, Existing File Name, New File Name,
Compliance Ruleset, BDL mod Table #1 Rec ID,
BDL mod Table #2 Rec ID, ..., Wizard -> BDL,
Perform Analysis, Program Output
Proc Rec, Existing File Name, New File Name,
Compliance Ruleset, BDL mod Table #1 Rec ID,
BDL mod Table #2 Rec ID, ..., Wizard -> BDL,
Perform Analysis, Program Output
End of File Flag
```

Figure 5. Explanation of fields in the example of batch directives file

Each field in the files presented above is delimited by commas. The explanation of these fields is given in the greyed text box below, with each field from the above table corresponding to the same field in the table below in green font color.

In the first row, the number "1" defines the Batch Version. Since the version is not bigger than 1, no further definition is required for the Batch Type.

In the second row, the number "0" defines the number of Project/Wizard/Energy Efficiency Measures modification tables. Following this number of tables is an equivalent number of strings that refer to the name of these modification tables. There are no strings written since there are no data modification tables of that type. If there were tables of this type, one could find the strings "*tbl_BDBLocations*" and "*tbl_BDBWeatherFile*". These strings would refer to the modification tables that contain respectively the geographical locations and the weather files that would be used to generate the new building models.

In the third row, the number "3" defines the number of BDL modification tables. Following this number of tables is an equivalent number of strings that refer to the name of these modification tables. The strings "*tbl_ChillerOversize*", "*tbl_BoilerOversize*", "*tbl_CTOversize*" refer to the modification tables that contain the oversizing factors of the chiller, boiler and cooling tower, respectively, that were used to generate the new building models.

In the fourth row, the number "1" defines the process record flag that identifies that this simulation record is to be performed indeed. The opposite would be the number "0" that would cause the simulation to skip the current simulation record and proceed to the next row. Following this flag are the file locations of the base case model used to generate the new models and where the new models will be saved represented respectively by the strings "*C:\SampleBaseProject*" and "*C:\SampleModifiedProject-A*". Next, a string defines the compliance ruleset that is to be used in the simulation. In this case, the "*unchanged*" string defines that no compliance analysis is to be performed, since it's not within the scope of this example or within the scope of the present research. After this, the record IDs of the data modification tables is defined, that is, which row of these data modification tables is to be used to generate the new model in question. The numbers "*2*", "*4*", "*3*" instruct the DOE engine to capture the second, fourth, and third rows of the data modification tables to generate the new model. Each one of these numbers refer to a different data modification table. The order used is the same order passed on in the second and third rows of this file, that is, in this example, the data modification tables that will be used as sources are respectively "*tbl_ChillerOversize*", "*tbl_BoilerOversize*", and "*tbl_CTOversize*". The following flag refers to how the program edits the input file with respect to the options previously edited in the Wizard. The number "0" instructs DOE only to modify the base case model parameters if these parameters have already been edited using the Wizard within eQUEST. Then, the string "*Simulate*" defines which type of analysis is to be performed, amongst the options of performing a regular simulation, a building Wizard simulation, an Energy Efficiency Measure run or a compliance analysis. The last field contains the string "*CumCSV*" that defines the program output. In this case, this string refers to a cumulative CSV approach, in which the results of each new simulation run are added after the last row of the output file without rewriting previously recorded simulation outputs.

The fifth row contains the same sort of instructions as the ones used in the fourth row. In this row, a new simulation run with different characteristics are defined. The user can subsequently add new rows to generate as many variation cases as he wishes.

The sixth and last rows contains the number "-1" identifying that this is the end of the file.

Tables of Data Modifications Creation Process

The process of creating the tables of data modifications file is comprised of the following subprocesses: (1) Define batch mods column keys, and (2) Define batch mods data.

The first sub-process in the process flow of the tables of data modifications file creation is the **Define Batch Mods Column Keys**. In this sub-process, numeric identifiers that indicate which columns of the current data table contain the parameters to be passed on to create the variant models are defined.

After this step, the property and the component being changed are also defined, separated by a colon (e.g. *Component:Property*). It can be either a property from the .pd2 file, such as the *WeatherFile* property under the *Proj* (Project) component, or a property from the .inp file, such as the *Type* property under the *Boiler* component. Another possibility is to use this field to modify global parameters, either symbolic or numeric instead of changing a specific property of a specific component of the model input files. A rulelist can also be used to evaluate which properties are to be changed.

Next, it must be defined how components are referenced. Components to be changed can be referenced either by their component name, identified by the flag "1" or by index, representing the order of occurrence, identified by the flag "0". In the column immediately after this one, the user informs the component name (e.g. *Boiler1, Boiler2, etc.)* or order of occurrence (e.g. *1, 2, etc.)*. Lastly, the type of data being modified is informed among four options: integer, float, string or symbol. The procedure is repeated, and rows are added to the file until the user no longer wishes to change properties under that specific component or global parameter.

The second and last sub-process is the **Define Batch Mods Data**. In the previous sub-process, instructions regarding to the type of data being modified were passed. In this sub-process, the actual numeric or text data to which the parameters will be modified to are defined. First, an index is provided for each entry. This index starts with the number "1" for the first row and each next row added will be the index of the previous row increased by one unit. Within each row, in the subsequent columns, the desired values for the property or properties to be changed are defined. The number of columns filled will be equivalent to the number properties to be changed defined in the Define Batch Mods Column Keys sub-process.

Example of Table of Data Modification File

An example of a table of data modification CSV file generated according to the instructions presented in the previous section is reproduced in its entirety in the greyed text box below. This example is a shortened and simplified version of one of the files used in the case study.

```
2,PARAMETER:Numeric,1,ChillerOversize,Float
-1
1,1.000
2,1.100
3,1.200
4,1.300
5,1.400
-1
```

Figure 6. Example of Table of Data Modification File

```
Table Column, Wiz Rulelist OR COM:KEY[#], By
Name or Index, Component/Variable Name or
Index, Type of Data
End of Batch Mods - Column Key
Index;Value
Index;Value
Index;Value
Index;Value
Index;Value
End of File Flag
```

Figure 7. Explanation of fields in the example of table of data modification file

Each field in the file presented above is delimited by commas. The explanation of these fields is given in the greyed text box below, with each field from the above table corresponding to the same field in the table below in green font color.

In the first row, the number “2” defines that this property will draw from the values contained in the second column of the rows that follow the end of the column key section of the current file. Next, the string “PARAMETER:Numeric” defines that a numeric value property of a parameter component will be modified to the values present in the next section of the file. The number “1” defines that the component to be changed will be searched by its name, rather than by its order of occurrence (index). After this, the string “ChillerOversize” defines the name of the component that is to be searched and modified. Lastly, the string “Float” defines that the data type of the “Type” property is a floating value. This step can be repeated, and rows are added to the file until the user no longer wishes to change properties from a specific component or global parameter.

In the second row, the flag “-1” defines the end of the batch modification column keys. This begins a new section of the file that contains the values that will be assigned to the properties listed in the preceding rows.

In the third row, the number “1” defines this row as the first entry in the index of available rows containing values to be passed on to the properties. In the second column of the fourth row, the value “1.000” defines the numeric value property of the “ChillerOversize” parameter component. The second column (column 2) of this row contains the same information as defined by the number “2” in the batch mods column key section, in the first row of the file. Subsequent columns could be added to change properties defined in the first section of this file.

In the fourth row, the same logic of construction used in the preceding row is used. The index number is increased by one unit and subsequent values for the “ChillerOversize” parameter are defined. This step can be repeated, and rows are added to the file until the user no longer wishes to change properties under that specific component or global parameter.

In the eighth row, the flag “-1” defines the end of the table of data modifications file.

Data Extraction and Compilation

The process of data extraction and compilation is comprised of the following subprocesses: (1) List folder contents with .SIM extension; (2) Create copies of the .SIM files and save the new copies as .txt files; (3) Create a table by reading the contents of the .txt files; (4) Delete rows containing warning/error reports (ATTN) in each cell of the table; (5) Create variables and assign values to them collected from the table; (6) Convert the string variables to double precision, and (7) Calculate derived check figures and KPIs from the double precision variables. This process is performed by a MATLAB code that is executed after the batch simulations have been run and all the output files have been generated.

The general approach of the MATLAB code is to import the information contained in the .SIM files of the simulations (using the *readtable* command) and use the imported values to calculate the desired parameters. The .SIM files contain all the output reports generated by the DOE program, including error and warning reports and pre-programmed hourly reports. Within the selected output folder, after the batch simulations are run, the user can find one .SIM file for each one of the variant models. One of the aspects that make working with the .SIM files so convenient is that they are structed using delimited-separated values, using as the delimiter the space character and are able to be opened using any text editor.

2.5. Sensitivity Analysis

In sensitivity analysis, the sensitivity of an output variable is studied as a function of the change of an or a set of input variables, explored around a base (reference) point. The regression method is the most widely used method in building energy analysis due to this method being easy to implement, fast to compute and easy to interpret the results (Tian, 2013). This paper uses a matrix form implementation of the regression method, where the Sensitivity Coefficients (SCs) that shows the impact of an input variable X on a KPI Y is given in Eq. 1 (Ryan et al., 2018):

$$SC = (X^T X)^{-1} X^T Y \tag{1}$$

Where X is a matrix composed of the inputs X_n for the sensitivity analysis and Y is the vector of the KPIs, each distinct one denoted by Y_n. Both X and Y were normalized over the values for the baseline case in Eq. 2 and Eq. 3.

$$X=(X_n-X_{base})/X_{base} \tag{2}$$
$$Y=(Y_n-Y_{base})/Y_{base} \tag{3}$$

Through the normalization all variations are represented as SC in percent changes over the base case; this allows the comparison of input variables that have distinct units.

3. RESULTS AND DISCUSSION

In this section, the results of the simulations and the sensitivity analysis are presented. A total of 75 model runs were performed, with a total elapsed time for the simulations of 23 minutes. The data extraction and compilation performed in MATLAB and the calculation of the sensitivity coefficients (post-processing) added 58 seconds to the total elapsed time. The average time per model run including simulation and post-processing is 19.2 seconds.

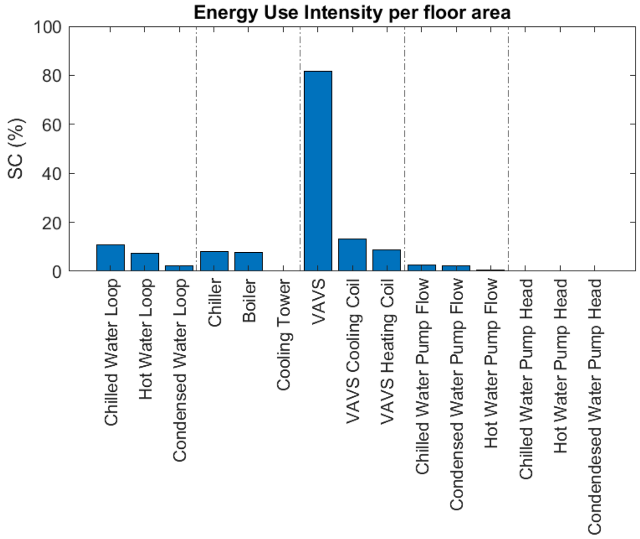

Figure 8. Sensitivity Coefficients correlating the EUI to the oversizing factors

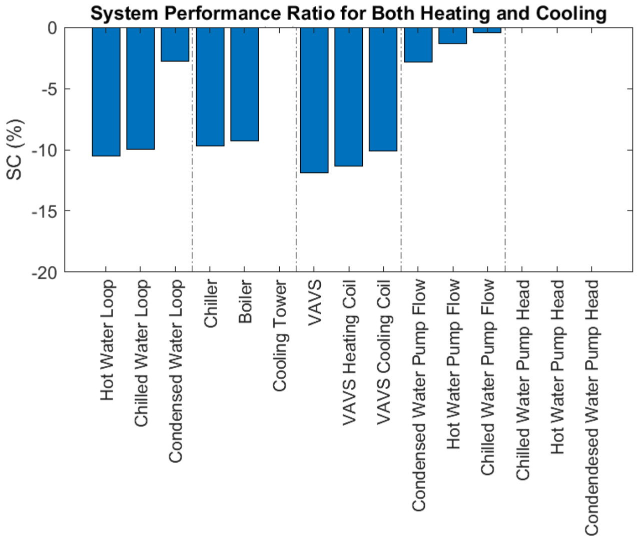

Figure 9. Sensitivity Coefficients correlating the System Performance Ratio to the oversizing factors

In Figure 8, the sensitivity coefficients correlating the EUI (including electric energy and natural gas energy) to the oversizing factors are shown in a bar plot, with vertical lines separating the category of the input variables. The secondary equipment category presented the most impact on the EUI, with the oversizing of the VAV System by 100% (doubling the size) causing an increase on the energy consumption of ~80%. Individually oversizing the cooling coil and the heating coil by the same amount resulted in an increase of ~13% and ~9%, respectively.

The second most impactful change is oversizing the water loops. This oversizing refers to the safety factor applied in the auto-sizing of the primary equipment based on secondary loop loads. Oversizing the entire chilled water loop (including all primary equipment connected to it) by 100% (doubling the size) resulted in an increase on the energy consumption of ~11%. Oversizing the hot water loop and the condensed water loop by the same amount resulted in an increase of ~7% and ~2%, respectively.

Next, with similar values to the ones presented by the loops, the oversizing of primary equipment (applied only to the equipment and not to the whole loop) is the most impactful factor. Oversizing the chiller by 100% (doubling the size) resulted in an increase on the energy consumption of ~8%. Oversizing the boiler by the same amount resulted in an increase of ~8% and oversizing the cooling tower resulted in a negligible effect.

Lastly, oversizing the pump flows presented the smallest effect on the EUI. Oversizing the chilled water pump flow by 100% (doubling the size) resulted in an increase on the energy consumption of ~3%. Oversizing the condensed water pump flow by the same amount resulted in an increase of ~2% and oversizing the hot water pump flow resulted in a negligible effect. Oversizing the pump heads instead of the pump flow resulted in a negligible effect on the EUI.

In Figure 9, the impact of the oversizing factors on the efficiency of both heating and cooling systems together is presented. The most impactful oversizing is verified on the secondary systems. Oversizing the VAV system by 100% incurred in a reduction of ~12% in efficiency, followed by ~11% and ~10% when the cooling coil and heating coil were oversized, respectively. Following the secondary system, oversizing the water loops presented the highest impact. Oversizing the hot water loop by 100% reduced the efficiency of the system by ~11%, followed by a reduction of ~10%, ~3% by oversizing the chilled water and condensed water loop respectively. Third, oversizing primary equipment presented a reduction of ~10% and ~9% in the efficiency when the chiller and boiler were oversized, respectively. Oversizing the cooling tower presented a negligible effect on the system performance. Lastly, with a smaller impact was the oversizing of the pump flows. Oversizing the condensed water pump flow, hot water pump flow and chilled water pump flow, respectively incurred in a reduction in the system performance of ~3%, ~1% and ~1%. Oversizing the pump heads presented a negligible impact on the system performance.

Using eQUEST as the simulation program proved to be time efficient, considering the architecture of the model. The program was able to simulate a three-storey building with 21 spaces in little less than 20 seconds per model run. Using MATLAB as the post-processing program for data extraction, compilation and sensitivity analysis complemented eQUEST's functionalities with little increase in time (~5% increase to the total simulation time).

Based on the results of the case study, a strong relationship between oversizing and both increased energy consumption and decreased system efficiency can be drawn. This relationship is likely due to two factors: (1) the oversized systems inevitably addressing higher thermal loads due to increased water and air flow and reduced operating temperatures, and (2) decrease in equipment efficiency due to lower part-load values. Oversizing beyond the systems needs was not beneficial for any of the input variables and KPIs chosen. Careful attention should be paid to the oversizing factors during design phase as not to cause these negative effects on the HVAC system. Although oversizing is highly connected to these negative effects, engineer discretion should be considered when selecting the oversizing factors, especially if changes are being made to an existing system, where the equipment has degraded performance and is unable to operate in design conditions.

In regard to the impact of oversizing factors, the secondary equipment presented the most impact. Among the selected input variables, oversizing the entire VAV system or the heating or cooling coils resulted in a substantial increase in energy consumption and a substantial decrease in system performance. The energy consumption was less affected by oversizing the loops and primary equipment, while the system performance was still substantially affected by the loops and primary equipment. Oversizing the pump flows and pump heads did not impact considerably the energy consumption or the system performance.

Comparing the impacts of oversizing on the EUI and on the system performance, one would expect that the impact of oversizing primary equipment would be bigger. The first factor that changes this balance is the availability of different energy sources supplying the system (electricity and natural gas). In the base case scenario, approximately half of the EUI comes from natural gas, which is directly connected to heating only. In addition, the proportion of heating to cooling loads can influence how changes in efficiency translate to changes in energy use. In a heating dominated climate, such as the one of Montréal, the energy performance of the system is strongly connected to the heating equipment. Lastly, even though the setpoint temperatures for all the simulated models are the same, the actual operating temperatures are a function of the system design and operation. Due to this effect, the elevated energy use related to oversizing the system is not only related to a decreased in efficiency, but also to the increased thermal loads being addressed by the system.

In this study, standard sizing procedures used by eQUEST were employed. In addition, control and automation procedures are implemented automatically by eQUEST in order to attempt to correctly operate the oversized systems. In practice, it is likely that oversizing of single equipment while keeping the same design for the rest of the system, particularly in control and automation, would result in relevant changes to how the system functions. These changes could affect system performance in ways that are not predicted in the scope of this study. An example of these changes could be the oversizing of the pump heads. In practice, with constant speed pumps, the excess pressure would be addressed by adjusting the balancing and control valves, which would increase the energy use of the system. With variable speed pumps, the pump power would be modulated according to the pressure setpoint and potential energy savings could be achieved. The aforementioned procedures are explained in detail in the DOE-2 Engineers and Reference manuals.

Lastly, it is observed from the simulation results that oversizing the condensed water loop resulted in a negative impact, probably due to the oversizing of all equipment connected to the loop. In contrast, oversizing the cooling tower individually didn't result in the same negative effect. Oversizing the cooling tower usually results in smaller condensed water temperatures, which is beneficial to the chiller performance. These beneficial effects are likely counterbalanced by increased energy spending in the cooling tower fan. The net effect on energy consumption and system performance is negligible.

4. CONCLUSIONS AND FUTURE WORK

BEM/BPS programs have been used extensively in in the design of buildings by providing load calculations and energy analysis. In these applications, simulation programs are commonly used to create a baseline building energy model, or a series of models, which serve as a reference to compare variants incorporating changes reflecting new or proposed scenarios. Batches of simulations are conducted over a range of climate zones, including these models, or series of models, developed to be statistically relevant building representations. These results of the batches of simulation can be compared using energy or cost performance KPIs and conclusions can be drawn to further the objective of reducing energy consumption or financial expenditure.

Due to the nature of the batch simulation process, hundreds or thousands of models representing specific building types and weather scenarios are generated. The need arises for a tool that can perform batch simulations with a user-friendly interface and that is capable of collecting and presenting energy performance results that can be interpreted intuitively through the comparison of KPIs. This research identified a gap in the literature as to how to perform batch simulation by using the eQUEST program. A detailed description of the batch processing feature was provided, with examples of input files generated in a spreadsheet file and the steps necessary to run the batch simulations. This serves as contribution to the documentation of the eQUEST program and elevates the usage of the program to new applications in a reliable and easy-to-use way.

In terms of the programs employed in the case study, using eQUEST as the simulation program proved to be time efficient, considering the complex architecture of the model. Using MATLAB as the post-processing program for data extraction, compilation and sensitivity analysis complemented eQUEST's functionalities with little increase in simulation time. Based on the results of the case study, a strong relationship between oversizing and both increased energy consumption and decreased system efficiency can be drawn. Careful attention should be paid to the oversizing factors during design phase as not to cause these negative effects on the HVAC system. Although oversizing is highly connected to these negative effects, engineer discretion should be considered when selecting the oversizing factors.

In regard to the impact of oversizing factors, the secondary equipment presented the most impact. Among the selected input variables, oversizing the entire VAV system or the heating or cooling coils resulted in a substantial increase in energy consumption and a substantial decrease in system performance. The energy consumption was less affected by oversizing the loops and primary equipment, while the system performance was still substantially affected by the loops and primary equipment. Oversizing the pump flows and pump heads did not impact considerably the energy consumption or the system performance.

In order to generalize the conclusions drawn in this study, performing the same oversizing factor studies in a range of climates, including more building archetypes would be necessary. A complete picture could be obtained by adding different HVAC system architectures to these building models.

5. REFERENCES

Clarke, J. A., & Hensen, J. L. M. (2015). Integrated building performance simulation: Progress, prospects and requirements. *Building and Environment*, *91*, 294–306. https://doi.org/10.1016/j.buildenv.2015.04.002

Hong, T., Chou, S. ., & Bong, T. . (2000). Building simulation: an overview of developments and information sources. *Building and Environment*, *35*(4), 347–361. https://doi.org/10.1016/S0360-1323(99)00023-2

Lerond, J. (2017). *eQUEST Batch Processing Documentation*. https://lymerej.gitbooks.io/equest-batch-processing-documentation/content/functioning.html

Madison, K. (2012). *Large Batch DOE-2 Modeling With eQUEST*. IBPSA Seattle. https://docs.google.com/open?id=0B2_lyY0pvJmVZHgwd3lDR3VvU0E

Natural Resources Canada. (2017). *National Energy Code of Canada for Buildings 2017*. https://nrc.canada.ca/en/certifications-evaluations-standards/codes-canada/codes-canada-publications/national-energy-code-canada-buildings-2017

Numeral Logics. (1999). *Canadian Weather for Energy Calculations, Users Manual and CD-ROM*. Environment Canada.

Rallapalli, H. S. (2010). *A Comparison of EnergyPlus and eQUEST Whole Building Energy Simulation Results for a Medium Sized Office Building*. M.Sc. Thesis, Arizona State University, Tempe, AZ, U.S.A.

Ryan, E., Wild, O., Voulgarakis, A., & Lee, L. (2018). Fast sensitivity analysis methods for computationally expensive models with multi-dimensional output. *Geoscientific Model Development*, *11*(8), 3131–3146. https://doi.org/10.5194/gmd-11-3131-2018

Tian, W. (2013). A review of sensitivity analysis methods in building energy analysis. *Renewable and Sustainable Energy Reviews*, *20*, 411–419. https://doi.org/10.1016/j.rser.2012.12.014

Tian, W., Liu, Y., Zuo, J., Yin, B., Sun, Y., Fu, X., & Meng, X. (2017). Building Energy Assessment Based on a Sequential Sensitivity Analysis Approach. *Procedia Engineering*, *205*, 1042–1048. https://doi.org/10.1016/j.proeng.2017.10.168