



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL

**MASSIVE 3D MODELS AND PHYSICAL DATA  
FOR BUILDING SIMULATION AT THE URBAN  
SCALE: A FOCUS ON GENEVA AND CLIMATE  
CHANGE SCENARIOS**

Autor: José Ramón Vázquez Canteli  
Director: Jérôme Henri Kämpf

Madrid  
Julio de 2016

José Ramón  
Vázquez  
Canteli

**MASSIVE 3D MODELS AND PHYSICAL DATA FOR BUILDING SIMULATION AT THE URBAN  
SCALE: A FOCUS ON GENEVA AND CLIMATE CHANGE SCENARIOS**



## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO**

### ***1º. Declaración de la autoría y acreditación de la misma.***

El autor D. **JOSÉ RAMON VÁZQUEZ CANTELI** DECLARA ser el titular de los derechos de propiedad intelectual de la obra: “**MASSIVE 3D MODELS AND PHYSICAL DATA FOR BUILDING SIMULATION AT THE URBAN SCALE: A FOCUS ON GENEVA AND CLIMATE CHANGE SCENARIOS**”, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### ***2º. Objeto y fines de la cesión.***

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### ***3º. Condiciones de la cesión y acceso***

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### ***4º. Derechos del autor.***

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

### ***5º. Deberes del autor.***

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho

- de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
  - c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
  - d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

**6º. Fines y funcionamiento del Repositorio Institucional.**

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 18 de Julio de 2016.

**ACEPTA**



Fdo JOSÉ RAMÓN VÁZQUEZ CANTELI

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

.....  
Massive 3D models and physical data for building simulation at the urban scale

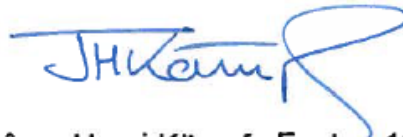
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2015-2016 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.



Fdo.: José Ramón Vázquez Canteli      Fecha: 18/ 07/ 2016

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Dr Jérôme Henri Kämpf      Fecha: 18/ 07/ 2016

Vº Bº del Coordinador de Proyectos

Fdo.: José Ignacio Linares Hurtado

Fecha: 19/ 07/ 2016





## **MODELOS 3D MASIVOS Y DATOS FÍSICOS PARA SIMULACIÓN DE EDIFICIOS A ESCALA URBANA: UN ESTUDIO SOBRE GINEBRA CON ESCENARIOS DE CAMBIO CLIMÁTICO**

**Autor: Vázquez Canteli, José Ramón.**

Director: Kämpf, Jérôme Henri.

Entidad colaboradora: École Polytechnique Fédérale de Lausanne (EPFL)

### **RESUMEN DEL PROYECTO**

En un periodo en el que los gobiernos persiguen objetivos de eficiencia energética cada vez más ambiciosos [IEA12], el campo de la simulación energética en edificios a escala urbana, combinada con una buena disponibilidad de datos de las características físicas, puede ayudar a cumplir con estos objetivos [KEIR12]. Este documento pretende, en primer lugar, describir la metodología para realizar estos modelos físicos en 3D de la ciudad de Ginebra. En segundo lugar, pretende analizar el potencial de ahorro energético de diferentes medidas de reforma de edificios así como proporcionar información sobre la cantidad de energía eléctrica que podría generarse mediante la instalación de más paneles fotovoltaicos. Además, en este estudio se analiza la firma energética de los edificios con el objetivo de comprender mejor el comportamiento de la demanda de calor en los distintos tipos de edificios y en función de su nivel de aislamiento. En lo relativo al control de los niveles de confort, se estudia el porcentaje estimado de personas satisfechas en función de la potencia nominal del equipo suministrador de calor y del volumen de los tanques de almacenamiento de calor, teniendo en consideración los respectivos costes variables y de inversión. Finalmente, se propone un algoritmo de control mejorado de la temperatura del tanque de almacenamiento de calor para su implementación en la herramienta de simulación CitySim.

### **Metodología**

La base de datos, cuya estructura se muestra en la Figura 1, se completó con datos del Sistema de Información del Territorio de Ginebra (SITG) [SITG16] y del Registro Federal de Edificios y Viviendas (RegBL) [REGB16]. La información está organizada en varias tablas, de las que la tabla de edificios es la más relevante ya que incluye no sólo las geometrías de los edificios, sino también otras características físicas y sus números de identificación, denominados códigos EGID, que se emplean para establecer las asociaciones entre las distintas tablas. La tabla de edificios se carga con información procedente de otras tablas que contienen los datos originales de SITG y RegBL. Además, la tabla que contiene la información de los paneles fotovoltaicos apunta a la tabla de edificios. Una vez la base de datos está completa, la herramienta de simulación CitySim necesita leer los datos en formato

XML. Por tanto, los datos se extraen de la base de datos y se usan para crear un archivo XML, para lo que se emplea un programa en Java escrito durante este proyecto específicamente para realizar esta tarea. Una vez el modelo se ha simulado, CitySim devuelve un archivo de texto con los resultados, que se leen con otro programa en Java que los carga en la base de datos.

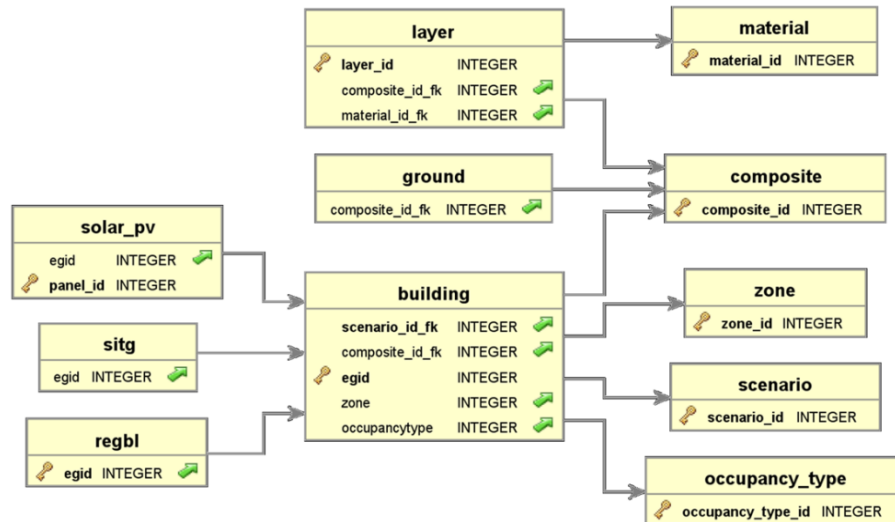


Figura 1: Estructura de la base de datos.

Para calibrar el modelo, los edificios son clasificados de forma que dicha clasificación sirva para representar las diferencias entre las características físicas de los distintos edificios. Una vez que los edificios están clasificados en función de su periodo de construcción y de su tipo de ocupación, las tasas de infiltración de aire y de ventilación se ajustan hasta minimizar la media del error en cada uno de los grupos de edificios.



Figura 2: Modelo físico en 3D en CitySim del distrito “La Jonction” en la ciudad de Ginebra



CitySim fue desarrollado en la EPFL [ROBI11] y permite estimar la demanda energética para calefacción, refrigeración e iluminación de cada edificio. Después de la simulación, CitySim devuelve un archivo de texto con todos los resultados para cada planta del edificio en escala horaria. Estos resultados se almacenan en la base de datos y están ligados a cada edificio mediante el código EGID. Esto permite analizar los resultados junto a las características físicas de los edificios. En este estudio se ha creado un modelo físico en 3D de un distrito de la ciudad de Ginebra que se muestra en la Figura 2.

## Resultados

El uso de modelos de regresión puede resultar útil para predecir la demanda energética en edificios en base a datos climáticos sin necesidad de recurrir a simulaciones. Por tanto, estos modelos de regresión pueden ayudar a predecir las demandas de calor, a gestionar el suministro de calor más eficientemente y a dimensionar el equipo más eficazmente en base a sus estimaciones. La irradiación global horizontal y la temperatura ambiente pueden medirse a escala urbana y usarse como variables regresoras. Sin embargo, la demanda de calor de los edificios que están altamente aislados tiene un nivel elevado de impredecibilidad que hace que estos modelos de regresión sean poco precisos.

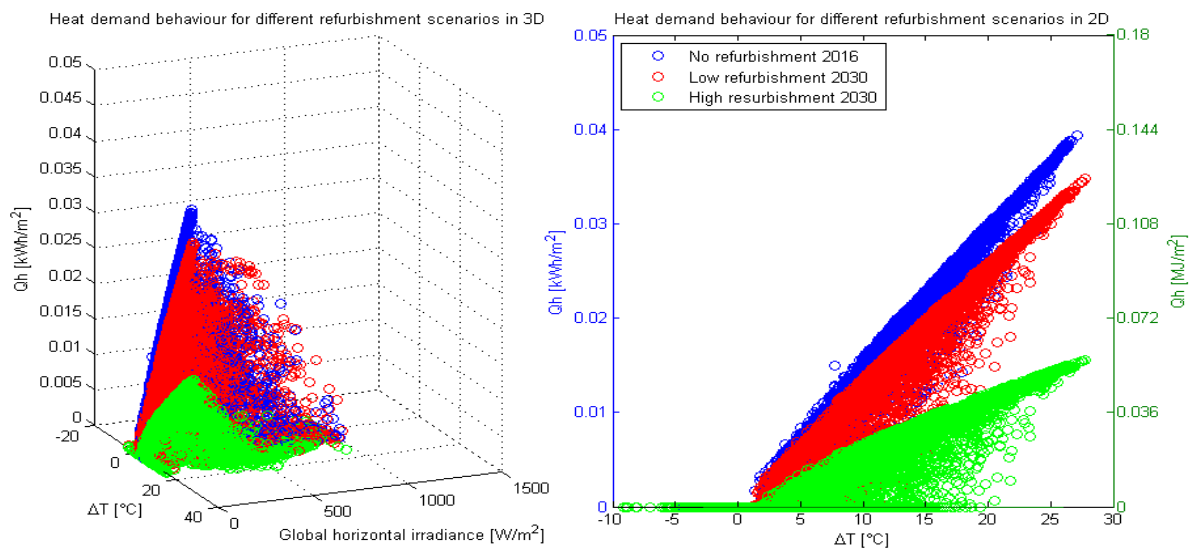


Figura 3: Firma energética de un edificio

La demanda de calor estimada para este distrito disminuye debido, en parte, a los efectos del cambio climático, y especialmente, cuando los edificios se aíslan bien, tal como la Figura 4 muestra. En todos los escenarios que se han tenido en cuenta, la demanda energética para refrigeración es demasiado baja como para justificar la implementación generalizada de sistemas de refrigeración en el distrito “La Jonction”. En el año 2050, aumentar los niveles de aislamiento y reducir las tasas de infiltración de aire disminuiría la demanda de frío en los edificios. Por tanto, un mayor aislamiento de estos edificios reduciría el efecto de la temperatura ambiente sobre la interior de los edificios más de lo que aumentaría el calentamiento producido por las mayores ganancias térmicas internas.

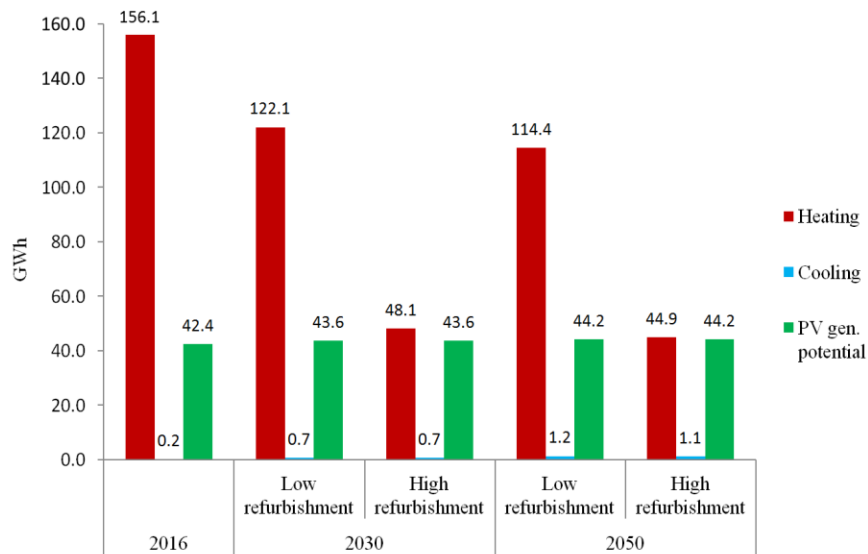


Figura 4: Evolución de la demanda de calor y del potencial fotovoltaico para distintos escenarios

El consumo neto de electricidad no se distribuye uniformemente como la Figura 5 muestra. Algunos edificios, localizados en la zona oeste del distrito, tienen el potencial de generar mucha más electricidad de la que necesitan. Las autoridades municipales pueden aprovechar este potencial y adoptar políticas que promuevan la instalación de sistemas centralizados de almacenamiento de energía cerca de estos edificios. Esta medida, combinada con las políticas existentes de reforma de edificios e instalación de paneles fotovoltaicos en los tejados, puede reducir mucho la dependencia de este distrito de las fuentes de energía no renovables.

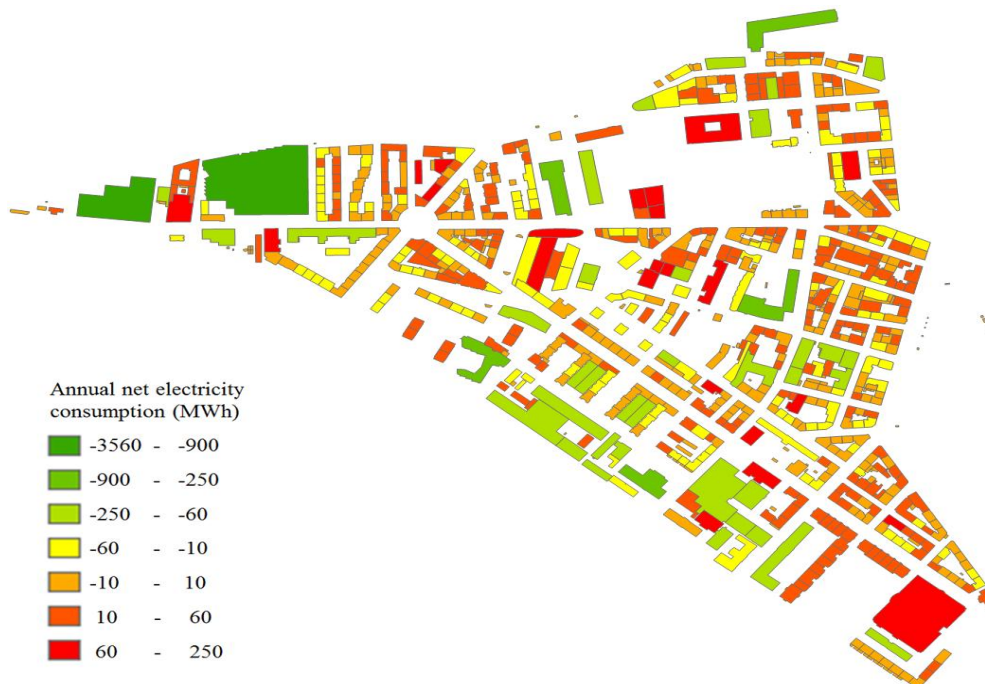


Figura 5: Mapa del consumo eléctrico anual neto estimado



## Conclusión

La simulación energética de edificios a escala urbana es una potente herramienta para estimar las necesidades energéticas y su potencial para generar energía. Las bases de datos PostgreSQL pueden almacenar las geometrías de los edificios. Una vez que el modelo físico en 3D se ha calibrado, se puede simular para obtener información valiosa sobre las necesidades energéticas o el potencial de generación de los edificios en distintos escenarios. Estos datos pueden almacenarse junto con las características de los edificios para un análisis más detallado. Para que estos modelos sean útiles, la disponibilidad de información precisa sobre las características de los edificios es crucial. Los resultados de la calibración muestran que es posible modelar los edificios residenciales con razonable precisión con la información disponible. Sin embargo, resulta más complicado lograr una precisión similar para otro tipo de edificios, especialmente aquellos en los cuales el comportamiento estocástico de los ocupantes es más impredecible y tiene un impacto mayor en la demanda energética.

La demanda y generación energética en el distrito de Ginebra fueron estudiadas para diferentes niveles de aislamiento de los edificios, teniendo en cuenta los efectos esperados debido al cambio climático en los años 2030 y 2050. Estos escenarios incluyen la implementación de paneles fotovoltaicos en cada tejado y el uso de bombas de calor. Estas medidas lograrían reducir drásticamente la dependencia energética de los edificios de fuentes no renovables e incrementar la generación eléctrica por encima de la demanda en el distrito. Sin embargo, este potencial superávit energético no se distribuye uniformemente. Algunos edificios tienen un potencial mucho mayor para la instalación de paneles fotovoltaicos, mientras que otros edificios con menos espacio disponible en sus tejados y un mayor volumen no serían capaces de producir más energía eléctrica de la que consumirían. Las autoridades municipales de Ginebra deben tener en cuenta estas asimetrías para adoptar políticas que promuevan una instalación eficaz de los paneles fotovoltaicos, de los sistemas de almacenamiento de energía y de las interconexiones en las redes de distribución energética.

Aprovechar al máximo aquellos edificios con un mayor potencial de generación eléctrica, instalar sistemas centralizados de almacenamiento energético, como tanques de agua, cerca de dichos edificios sería una posible solución. Una combinación de paneles fotovoltaicos y colectores solares podría ser un buen enfoque. El calor almacenado puede ser redistribuido por una red de distrito o convertido en electricidad para su distribución en el sistema eléctrico.

## Referencias

- [IEA12] Energy Policies of IEA Countries, Switzerland; International Energy Agency, 2012 Online. [www.iea.org/publications/freepublications/publication](http://www.iea.org/publications/freepublications/publication)
- [KEIR12] Keirstead, J., Jennings, M. & Sivakumar, A. “A review of urban energy system models: Approaches, challenges and opportunities”, Elsevier: London, 2012.
- [SITG16] SITG, Online (accessed: 10/07/2016): <http://ge.ch/sitg/>
- [REGB16] Federal Register of Buildings and Dwellings, Online (accessed: 10/07/2016): <http://www.bfs.admin.ch/bfs/portal/en/index/themen/00/05/blank/01.html>,
- [ROBI11] Robinson, D., “Computer modelling for sustainable urban design”, Earthscan: London, pp. 121, 2011.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL

---



## **MASSIVE 3D MODELS AND PHYSICAL DATA FOR BUILDING SIMULATION AT THE URBAN SCALE: A FOCUS ON GENEVA AND CLIMATE CHANGE SCENARIOS**

**Author: Vázquez Canteli, José Ramón.**

Director: Kämpf, Jérôme Henri.

Collaborator entity: École Polytechnique Fédérale de Lausanne (EPFL)

### **ABSTRACT OF THE PROJECT**

In a time when governments are pursuing increasingly ambitious energy efficiency goals [IEA12], building energy simulation at the urban scale, combined with the high availability of physical data, can help in performing this task [KEIR12]. This report intends, first, to describe the methodology to realize the geometrical 3D and physical model of the city of Geneva. Then, it analyzes the energy-saving potential of different refurbishment measures, and provides information about the energy generation that could be achieved thanks to the installation of PV panels. In addition, this study analyzes the energy signature of the buildings in an attempt to better understand the behavior of the heat demand in different types of buildings and levels of refurbishment. Regarding the control of the levels of comfort in a cost-efficient approach, the percentage of satisfied people was analyzed as a function of the power of the heating equipment and of the heat storage tanks' volume, taking into consideration their respective investment and variable costs. Finally, an improved control algorithm of the heat storage tank's temperature is proposed to be implemented in CitySim.

### **Methodology**

The database, whose structure is illustrated by fig. 1, was completed with data from the information system of the territory of Geneva (SITG) [SITG16] and from the Federal Register of Buildings and Dwellings (RegBL) [REGB16]. The information was organized in several tables, among which the table of buildings is the most important because it contains not only the geometries of the buildings and other physical characteristics, but also their identification numbers or EGIDs, which are used to establish associations among the different tables. This table of buildings is fed up with information from other tables that contain the original data from SITG and RegBL. Also the table that contains the information about the solar PV panels points to the table of buildings. Once the database is complete, the simulation tool CitySim needs to read the data in an XML format. Hence, the data must be extracted from the database and used to write this XML file. In this project, a java program was written and used to

perform this task. Once the model is simulated, CitySim outputs a text file with the results, which were read by another java program and put back into the database.

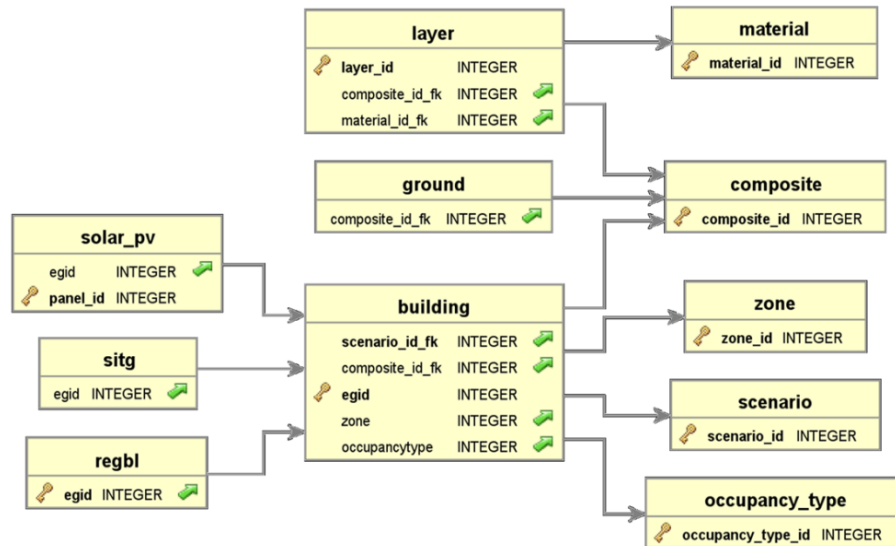


Figure 1: Structure of the database.

In order to calibrate the model, the buildings were classified in a meaningful way that best express the differences of the physical characteristics of the buildings. Once the buildings were classified as a function of their age of construction and occupancy type, the infiltration and ventilation rates were adjusted until the median error was minimized on each of the groups of buildings.



Figure 2: Geometrical 3D and physical model in CitySim of the Junction district in Geneva

CitySim was developed at EPFL [ROBI11], and computes an estimation of the energy demand for heating, cooling, and lighting of every building. After the simulation, CitySim returns a text file with all the results for every floor and building on an hourly basis. These results are put into the database and linked to each building by the EGID number. This allows analysis of the results in combination with the physical characteristics of the buildings. A geometrical 3D and physical model of one district of Geneva was created and is depicted by fig. 2.

## Results

The use of regression models can be useful to predict the energy demand of the buildings based on climatic data and without even running a simulation. Hence, these models can help in predicting the heat demand, managing the heat supply more efficiently, and sizing the equipment based on these estimations. The global horizontal irradiance and the average hourly temperature can be measured in an urban scale and used as regressors. However, the heat demand of highly insulated buildings has a high degree of unpredictability that makes the regression more difficult to perform as fig. 3 illustrates.

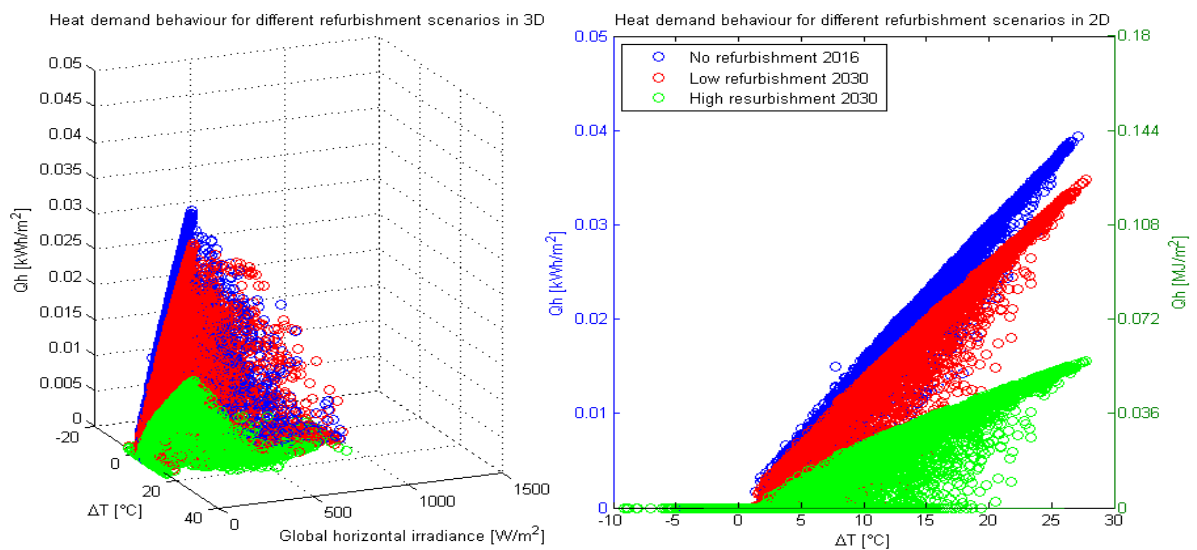


Figure 3: Energy signature of a building

The heat demand of this district would decrease due, in part, to the effects of climate change, while the cooling demand will increase, as shown in fig. 4. However, in all the scenarios considered, the predicted cooling demand would still be too low to justify the wide implementation of cooling systems in the Junciton district. In 2050, increasing the insulation of the walls and reducing the infiltration rates would result in lower cooling demands of the buildings. Therefore, the better insulation of these buildings would reduce the effect of the outdoor temperature on the indoor temperature, more than it would increase the heating due to the internal gains.

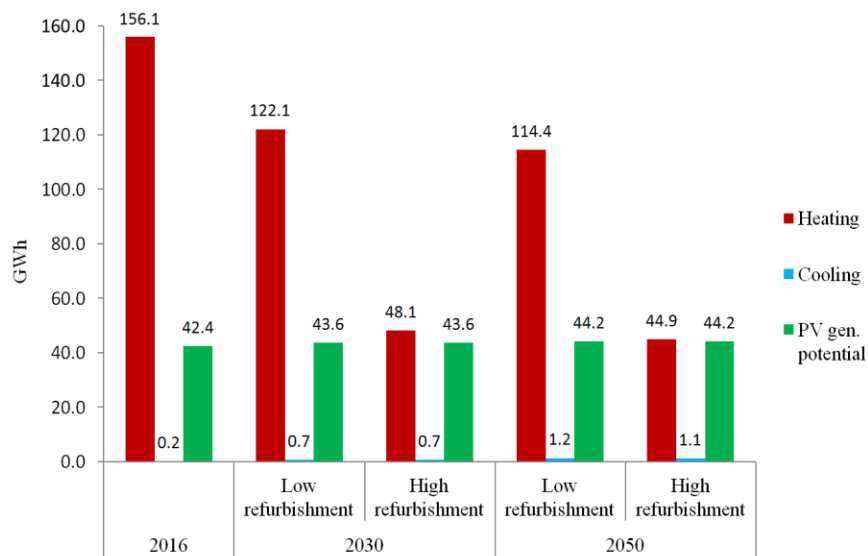


Figure 4: Evolution of the heat demand and the PV potential for different scenarios

The net electricity consumption is not uniformly distributed as fig. 5 shows. Some buildings, located on the west part of the district, have the potential to generate much more energy than they would need. Municipal authorities can take advantage of this potential, and adopt policies that promote the installation of centralized energy storage systems near these buildings. This measure, combined with the existing policies of building refurbishment and the installation of more PV panels on all the roofs, can drastically reduce the dependency of this district on non-renewable energy technologies.

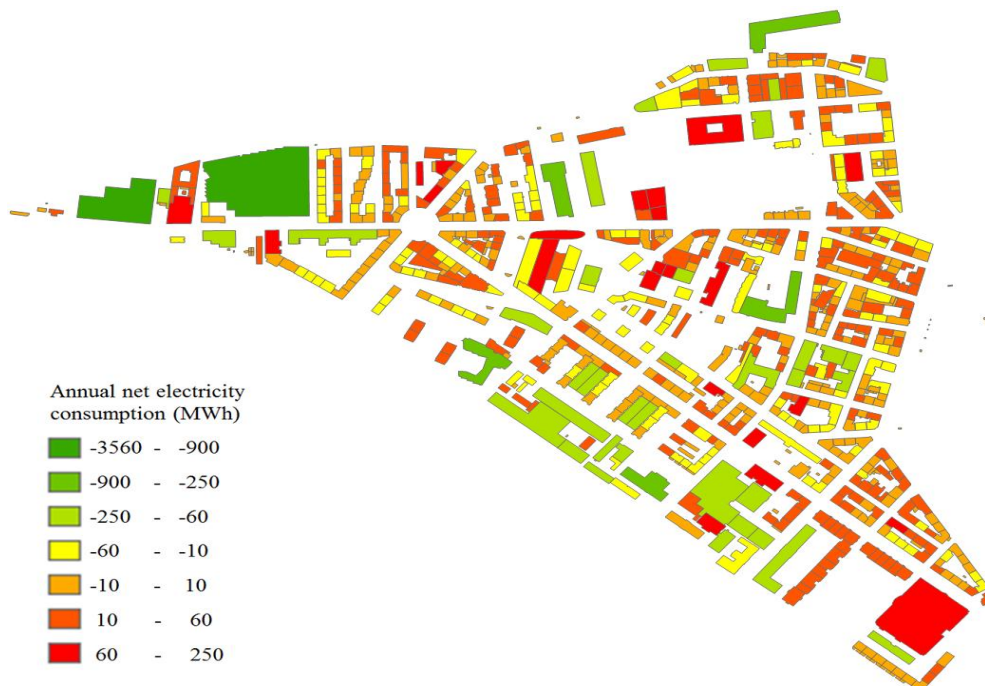


Figure 5: Map of the estimated annual net electricity consumption





## Conclusion

Building simulation at the urban scale is a powerful tool to simulate the energy needs and generation potential of cities. PostgreSQL databases can store all the geometries of the buildings and their physical characteristics. Once the geometrical 3D and physical model is realized and calibrated, it can be simulated to obtain valuable information about the energy needs or generation of the buildings in different pre-defined scenarios. These data can be stored in the database together with the physical characteristics of the buildings for further analysis. For these models to be useful, the availability of accurate information regarding the physical characteristics of the buildings and their geometries is crucial. The results of the calibration show how it was possible to model the residential buildings accurately with the information that was available. However, it is more difficult to achieve this accuracy for other types of buildings, especially for those in which the stochastic behavior of the occupants is more unpredictable and has a greater impact on the heat demand.

The energy demand and generation of a district in Geneva were studied for different refurbishment scenarios, taking into account the expected effects of climate change in 2030 and 2050. These scenarios included an ideal case with the implementation of PV panels on every roof, and the use of heat pumps in every building. These measures were proven to drastically reduce the energy demands of the buildings, while increasing their generation of electricity above the overall annual energy consumption of the district. However, this potential energy surplus is not uniformly distributed. Some buildings have a much greater potential for installing PV panels than others, while certain buildings with smaller rooftop areas and higher heights would not be able to generate more electricity than they would consume. Municipal authorities of Geneva must take into account these asymmetries to adopt policies that promote an efficient distribution of PV panels, energy storage systems, and adequate interconnections in the energy distribution network.

Taking maximum advantage of the buildings with the highest solar potential, and installing centralized energy storage systems, such as water heat tanks, near them would be a possible good solution. In this case, a combination of PV panels and solar heaters would be a good approach. The stored heat can be later redistributed through a district heating network or converted into electricity and distributed through the electrical power grid.

## References

- [IEA12] Energy Policies of IEA Countries, Switzerland; International Energy Agency, 2012 Online. [www.iea.org/publications/freepublications/publication](http://www.iea.org/publications/freepublications/publication)
- [KEIR12] Keirstead, J., Jennings, M. & Sivakumar, A. “A review of urban energy system models: Approaches, challenges and opportunities”, Elsevier: London, pp. 3851-3852, 2012.
- [SITG16] SITG, Online (accessed: 10/07/2016): <http://ge.ch/sitg/>
- [REGB16] Federal Register of Buildings and Dwellings, Online (accessed: 10/07/2016): <http://www.bfs.admin.ch/bfs/portal/en/index/themen/00/05/blank/01.html>,
- [ROBI11] Robinson, D., “Computer modelling for sustainable urban design”, Earthscan: London, pp. 121, 2011.





## Table of contents

1	Introduction .....	25
2	Methodology.....	27
2.1	Creation of the database .....	27
2.2	Creation of the geometrical 3D and physical model .....	29
2.3	Calibration of the model.....	33
3	Results and discussion.....	37
3.1	Solar PV integration .....	37
3.2	Energy signature of the buildings on an hourly scale.....	37
3.3	Energy signature of the buildings on a daily scale .....	43
3.4	Heat pump and heat storage tank dimensioning .....	47
3.5	An improved control algorithm for the heat pump – heat tank system .....	51
4	Conclusion.....	53
5	References.....	55
6	Acknowledgements .....	57
	Annex I: Conference paper .....	59
	Annex II: MATLAB code to read and display the heat demand behavior as a function of the temperature and the solar irradiance.....	69
	Annex III: MATLAB code to optimize the calibration using the Hypercube optimization method... 75	
	Annex IV : PostgreSQL code used to create the table that contains all the data of the buildings.....	83
	Annex V : Java code to implement the solar PV on the roofs of the buildings.....	101
	Annex VI : MATLAB code to evaluate the thermal comfort as a function of the heat tank volume and the HP electrical rated power.....	105
	Annex VII: MATLAB code to compare the different types of models.....	109





## Table of figures

Figure 1: Building data extracted from its data source SITG (left-side picture), processed by QGIS after the database is created, and the final model in CitySimPro (right-side picture).....	27
Figure 2: Geometrical 3D and physical model of the Junction district in Geneva. The PV panels were oriented using the rotation matrices shown in eqn. (2). .....	28
Figure 3: Model creation and simulation framework with interaction between CitySimPro and the Database.....	29
Figure 4: Geometrical 3D and physical models with different levels of detail (Model 3 on the left, and Model 1 and 2 on the right).....	30
Figure 5: Comparison of the annual heating demand of Model 2 and 3 with respect to Model 1 (the most detailed model) for 20 buildings. ....	30
Figure 6: Thermal behavior of the different models of two different buildings. The building represented by the graphs on the left column has greater solar gains, while the one represented by the right-side column is shadowed by the surrounding buildings. Each of the 24 hours represents the average of that same hour for the 365 days of the year. ....	32
Figure 7: Representation of the buildings whose behavior is explained in Figure 6. The building in yellow has greater solar gains, and the one in blue color is shadowed. ....	33
Figure 8: Optimization framework for the calibration process using the Hyper-cube optimization method programmed in MATLAB. ....	34
Figure 9: Conceptual 3D representation of the Hypercube optimization algorithm.....	35
Figure 10: Heat demand as a function of $\Delta T$ and the global horizontal irradiance for different refurbishment scenarios on an hourly basis. $R2_{adjusted} = [R2_{2016}, R2_{low}, R2_{high}] = [0.8928, 0.7848, 0.7346]$ .....	38
Figure 11: Heat demand as a function of $\Delta T$ and the global horizontal irradiance for different refurbishment scenarios on an hourly basis. $R2_{adjusted} = [R2_{2016}, R2_{low}, R2_{high}] = [0.9599, 0.9309, 0.8176]$ .....	39
Figure 12: Distribution of the adjusted R2 of the heat demand comparing various regressors on an hourly basis. ....	40
Figure 13: Distribution of the adjusted R2 of the heat demand with $\Delta T$ and the global horizontal irradiance as regressors. The effect of the occupants on these distributions is analyzed and compared between both graphs (the upper graph does not include the occupants, while graph at the bottom does). ....	41
Figure 14: Heat demand as a function of $\Delta T$ and the global horizontal irradiance for different refurbishment scenarios on an hourly basis and including the occupants. $R2_{adjusted} = [$	

	$R_{2016}, R_{2low}, R_{2high} ] = [0.8802, 0.7637, 0.7125]$ (figures above); $R_{2adjusted} = [R_{2016}, R_{2low}, R_{2high} ] = [0.3764, 0.3333, 0.3775]$ (figures at the bottom). .....	42
Figure 15:	Building whose thermal behavior is depicted by the graph at the bottom of Figure 14. ..	43
Figure 16:	Heat demand as a function of $\Delta T$ and the global horizontal irradiance for different refurbishment scenarios on a daily basis. $R_{2adjusted} = [ R_{2016}, R_{2low}, R_{2high} ] = [0.9911, 0.9890, 0.9570]$ .....	44
Figure 17:	Distribution of the adjusted R2 of the heat demand for different refurbishment scenarios and using different combinations of regressors on a daily basis. ....	45
Figure 18:	Distribution of the heat tank volume of the buildings for different refurbishment scenarios and assuming that the tank would be able to store all the heat that would be consumed in the day of maximal heat demand of its associated building.....	46
Figure 19:	Percentage of people satisfied as a function of the heat tank volume and the HP power. ....	48
Figure 20:	Price of comfort as a function of the heat tank size and the power of the heat pump. ....	49



## **Table of tables**

Tabla 1: Parameters used to model the behavior of the blinds.....28



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL

---





## 1 Introduction

Since the start of the industrial revolution there has been an increase in the carrying capacity of the land being exploited for agricultural purposes, which at the same time has led the global population to increase dramatically. This has enabled a great proportion of this population to reside in urban settlements. Therefore, the fraction of urban population has increased exponentially, to the extent that today there are 35 megacities accommodating more than 10 million people each. [ROBI11]. Given this dramatic increase of the global population, especially in cities, building simulation at the urban scale can provide municipal authorities with useful information to pursue more efficient and innovative solutions to this problem of the increasing energy demand in the urban territory.

Switzerland, for instance, is aiming to reduce its greenhouse emissions by 20% by 2020 while moving towards an energy mix in which the base load generation would be mainly satisfied by gas-fired power plants [IEA12]. Taking into account that the population living in the urban regions is increasing, these two strategy paths seem to be difficult to follow at the same time without decreasing the energy consumption drastically. Most of Switzerland's energy-related CO<sub>2</sub> emissions come from transportation and space heating.

Many solutions are already being implemented: more restrictive energy standards to reduce the heat demand in buildings, installation of more PV-T systems that supply DHW and electricity to the buildings, more effective HVAC controllers [PRIV10], and the integration of the electric vehicles as energy storage systems in the electric power grid. Cities' energy systems are increasing in complexity, and due to the increased insulation of the buildings, the heating and cooling demands are less dependent on the ambient temperature than ever before. Instead, the energy demand of the buildings depends more on stochastic behaviors such as solar irradiance and occupants' activities. Building simulation at the urban scale helps in estimating the heating and cooling demands accurately and taking into consideration all these stochastic phenomena. Furthermore, it is a useful tool for municipal authorities who want to compare different refurbishment and utility integration measures on a large scale [KARN09] [KEIR12]. Climate change estimations provided by the Meteonorm software are helpful in evaluating the expected effect of global warming on the heating and cooling needs of a city. Some models based on PostgreSQL databases and GIS have been used before to model other cities in Switzerland [PERE11] [PERE13].

This report intends, first, to describe the methodology to realize the geometrical 3D and physical model of the city of Geneva. Then, it analyzes the energy-saving potential of different refurbishment measures, and provides information about the energy generation that could be achieved thanks to the installation of PV panels. In addition, this study analyzes the energy signature of the buildings in an attempt to better understand the behavior of the heat demand in different types of buildings and levels of refurbishment. Regarding the control of the levels of comfort in a cost-efficient approach, the percentage of satisfied people was analyzed as a function of the power of the heating equipment and of the heat storage tanks' volume, taking into consideration their respective investment and variable costs. Finally, an improved control algorithm of the heat storage tank's temperature is proposed to be implemented in CitySim.

The conference paper "Massive 3D models and physical data for building simulation at the urban scale: a focus on Geneva and climate change scenarios" is based on this work and was published in the *11th International Conference on Urban Regeneration and Sustainability*. The full paper is included in the Annex I as part of this report, and some of the explanations will make reference to it.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL.

---

## 2 Methodology

The creation of the geometrical 3D and physical model is a process that needs large amounts of data of most of the physical characteristics of the buildings, climatic conditions, orography, and the occupants' behavior. These data was properly organized into a database, and some of the parameters were estimated based on typical values for each type of building. For these reason, it is important to classify the buildings in a way that such classification is meaningful and allows estimating these missing characteristics more easily. Finally, the model was calibrated with real data to correct the possible mistakes committed during the estimation process, and adjust the model to its real behavior. All this process is explained on the conference paper found in section 2 of the Annex I.

### 2.1 Creation of the database

As mentioned in the Annex I, the database was completed with data from the information system of the territory of Geneva (SITG) [SITG16] and from the Federal Register of Buildings and Dwellings (RegBL) [REGBL16]. The information was organized in several tables, among which the table of buildings is the most important because it contains not only the geometries of the buildings and other physical characteristics, but also their identification numbers or EGIDs, which are used to establish associations among the different tables. This table of buildings is fed up with information from other tables that contain the original data from SITG and RegBL. Also the table that contains the information about the solar PV panels points to the table of buildings.

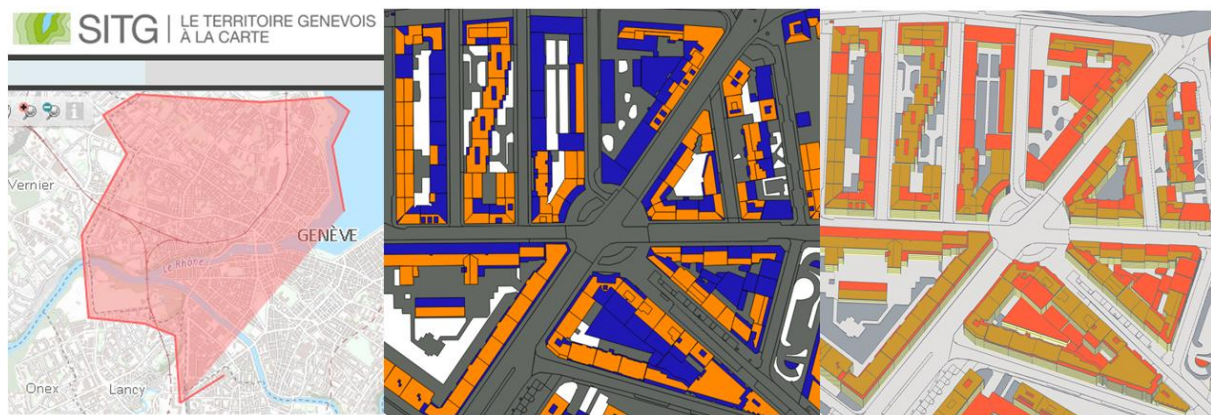


Figure 1: Building data extracted from its data source SITG (left-side picture), processed by QGIS after the database is created, and the final model in CitySimPro (right-side picture).

The walls of each building are made of a composite, which is at the same time comprised of several different layers, each of which is made of a specific material. All these relationships are well organized in tables and illustrated in Figure 1 of the Annex I. It is also worth pointing out that the buildings were classified by different occupancy types (i.e. residential, commercial) and in different zones, each of which represents a city district. The age of construction of the buildings, the occupancy types, and the zone are key elements on the estimation and calibration processes. The full PostgreSQL code used to create the table of buildings can be found in the Annex IV of this report.

Apart from all the characteristics mentioned in the Annex I, the model also takes into account the behavior of the blinds, which was modeled based on eqn. (1) and table 3, where  $f_u$  represents the unobstructed fraction of the glass, and  $I_0$  the solar irradiance.

Tabla 1: Parameters used to model the behavior of the blinds

Building type	$\lambda$ (m <sup>2</sup> /W)	Irradiance cut-off (W/m <sup>2</sup> )
Administrative	0.017	200
Others	0.017	400

$$f_u = \frac{1}{1+e^{\lambda(I_f-I_o)}} \quad (1)$$

The information about the solar PV potential of the roofs was obtained also from SITG. The geometrical 3D and physical model was implemented with a java code that can be found in the Annex V (only the section of the code that corresponds to the modeling of the PV panels is shown). As Figure 2 depicts, the code allows modeling the PV panels in 3D, including their orientation and inclination, although in the end only the 2D model was implemented due to the limitations of the data regarding the shape of the roofs.

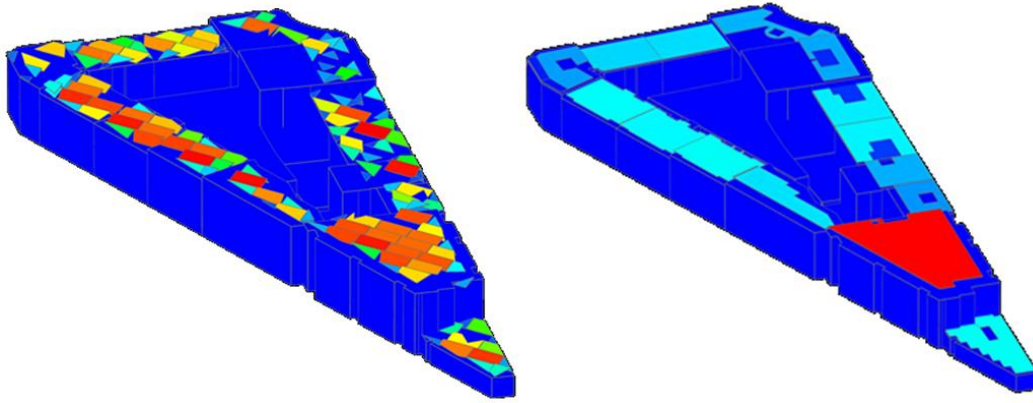


Figure 2: Geometrical 3D and physical model of the Junction district in Geneva. The PV panels were oriented using the rotation matrices shown in eqn. (2).

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{pmatrix} \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \quad (2)$$

$$x' = x_o - x_{centroid}$$

$$y' = y_o - y_{centroid}$$

$$z' = z_o - z_{centroid}$$

$$\text{orientation: } -\alpha$$

$$\text{inclination: } -\beta$$

## 2.2 Creation of the geometrical 3D and physical model

Once the database is complete, the simulation tool CitySim needs to read the data in an XML format. Hence, the data must be extracted from the database and used to write this XML file. In this project, a java program was written and used to perform this task. Once the model is simulated, CitySim outputs a text file with the results, which were read by another java program and put back into the database. Figure 3 illustrates this process.

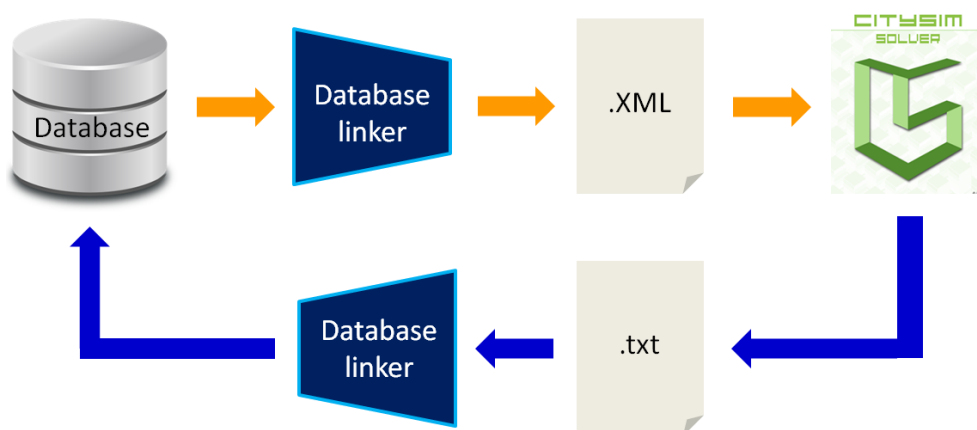


Figure 3: Model creation and simulation framework with interaction between CitySimPro and the Database.

The geometrical 3D and physical model is not only made of buildings but also of ground surfaces, which include roads, grass, asphalt, and even water of the river. These surfaces have different albedos and reflect the sunlight in different directions. They also absorb part of the solar irradiance, increasing their temperature and emitting long-wave irradiance, part of which is absorbed by the buildings. Figure 2 of Annex I depicts how this model looks like in CitySim.

The geometrical 3D and physical model should pursue a compromise between precision and the computational time it needs for the simulation. Therefore, it is possible to develop models with different levels of detail depending on our requirements. This section intends to analyze three different models, with different levels of detail. The models will be compared among them and not with the real measurements. The reason for this is that the calibration was done using the most simplified model, and this could lead to biased results, in which the heat demand of the model used in the calibration would be apparently the most similar to the real data without being necessarily the most precise model.

The simplest model to be analyzed does not include any floors, and each building has a single thermal zone, while each wall is made of a single surface. The left picture of Figure 4 illustrates this model, called Model 3.

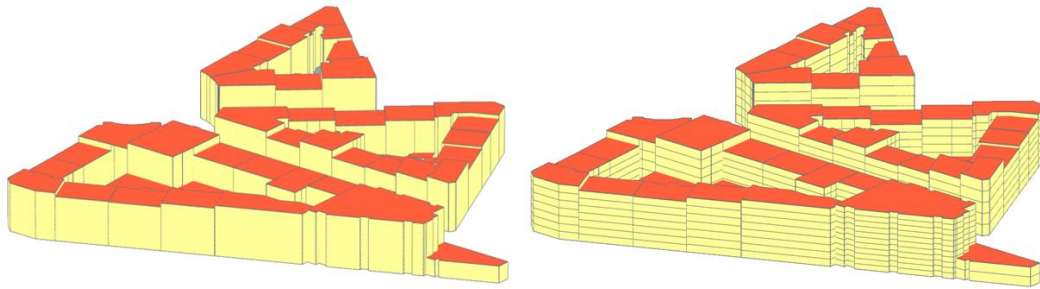


Figure 4: Geometrical 3D and physical models with different levels of detail (Model 3 on the left, and Model 1 and 2 on the right).

Models 1 and 2 are more detailed. Model 2 takes into consideration the divisions of the walls due to the different existing floors. However, in this model each building has a single thermal zone. On the other hand, Model 1 has as many thermal zones as floors, and each wall is also made of several surfaces that represent the different floors. Therefore, the thermal inertia of Model 2 and Model 3, which have a single thermal zone, is considerably lower than the inertia of Model 1 that has several thermal zones.

Figure 5 shows the results after simulating the 20 buildings depicted in Figure 4. In Figure 5, Model 2 and Model 3 are compared with Model 1, which is the most detailed one. Of these 20 buildings, there are 9 in which the heat demand is not the same as in Model 1. This difference varies from 0.5% to 8%.

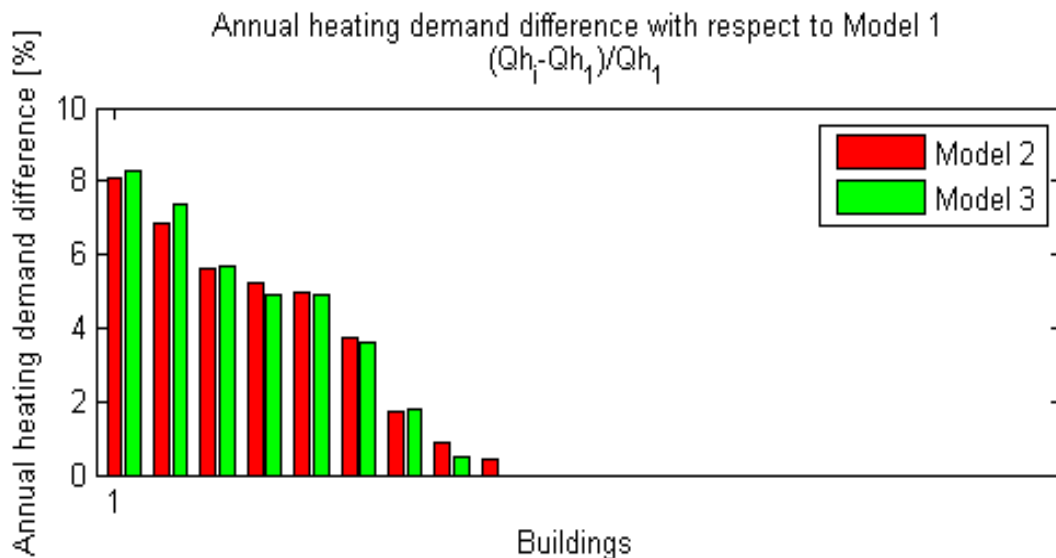


Figure 5: Comparison of the annual heating demand of Model 2 and 3 with respect to Model 1 (the most detailed model) for 20 buildings.



Model 2 and Model 3 have a very similar behavior, and since the difference between these two models with respect to Model 1 is their lower thermal inertia, it is worth analyzing the effect of their internal temperature ( $T_a$ ) on the heat demand ( $Q_h$ ). Figure 6 depicts the thermal behavior of two different buildings during the average 24 hours period. The graphs on the first column show a building whose solar gains are high, and so its  $T_a$  during the day is too. The building represented by the three graphs on the second column is very shadowed, and therefore its internal temperature barely changes within the day.

The high increase of  $T_a$  that can be observed in Model 2 and Model 3 due to their lower thermal inertia leads to an increase of the thermal losses through the walls and the windows. This behavior is expressed by eqn. (3). If the average yearly external temperature is  $10.9^\circ\text{C}$ , and the average  $T_a$  in Model 1 is  $21^\circ\text{C}$ , an increase of  $0.8^\circ\text{C}$  of  $T_a$  will lead to an 8% increase of the thermal losses of the building. And since the heat supplied must equal the heat losses, it also involves an 8% increase of the heat demand.

$$Q_{loss} = U_{wall} \cdot A_{wall} \cdot (T_a - T_{ext}) \quad (3)$$

This is not the case of the building whose thermal behavior is depicted in the graphs of the second column. The solar gains are low due to the shadowing effect of the surrounding buildings, which leads to a lower  $T_a$  increase and, therefore, lower thermal losses.

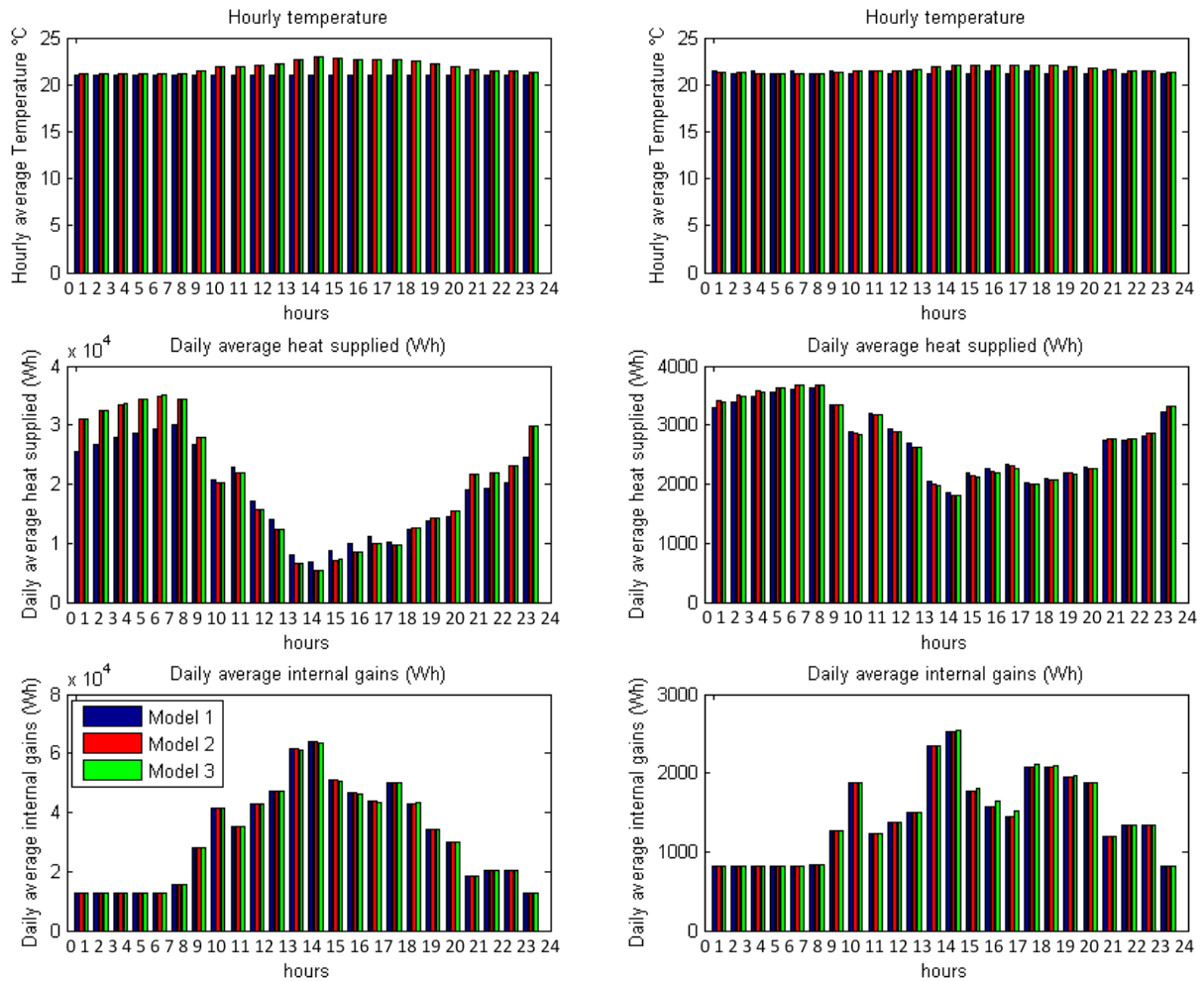


Figure 6: Thermal behavior of the different models of two different buildings. The building represented by the graphs on the left column has greater solar gains, while the one represented by the right-side column is shadowed by the surrounding buildings. Each of the 24 hours represents the average of that same hour for the 365 days of the year.

Figure 7 highlights the buildings whose behavior is depicted by Figure 6. The building in yellow has greater solar gains and  $T_a$  increment, while the building in blue color is shadowed by the surrounding buildings and its internal temperature barely changes among the three different models. Therefore, it is possible to conclude that the shadowing and the solar gains play a very important role in determining whether or not a building can be well simulated with a simplified model. Shadowed buildings can be modeled with a single thermal zone, while buildings that receive high solar irradiation should be modeled as multiple-zone buildings to take into consideration their higher thermal inertia and lower heat losses. Therefore, the computational time required to simulate a model of a district can be reduced by modeling in higher detail (with as many thermal zones as floors) the buildings that are very exposed to solar irradiation, and with lower detail (with a single thermal zone) those that are very shadowed by the surrounding buildings. The MATLAB code used to analyze the differences between the different models is included in Annex VII.



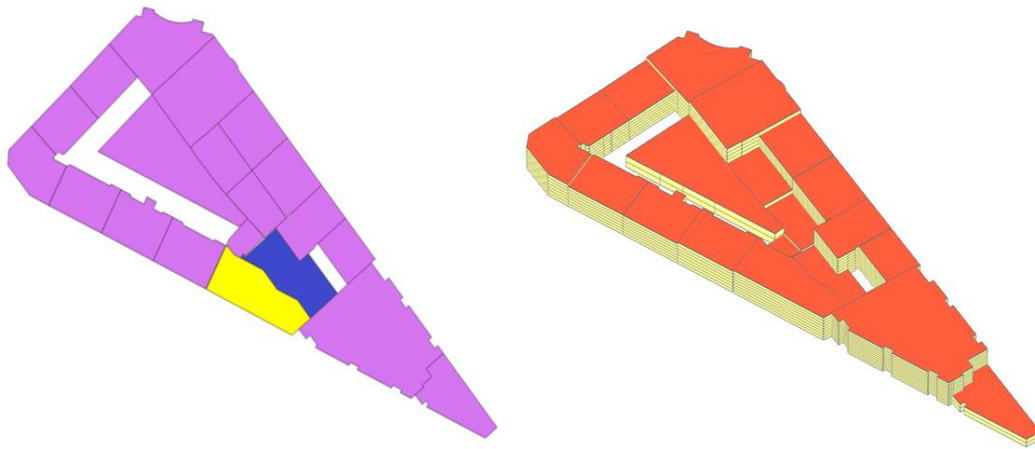


Figure 7: Representation of the buildings whose behavior is explained in Figure 6. The building in yellow has greater solar gains, and the one in blue color is shadowed.

### 2.3 Calibration of the model

In order to calibrate the model, the buildings were classified in a meaningful way that best express the differences of the physical characteristics of the buildings. Once the buildings were classified as a function of their age of construction and occupancy type, the infiltration and ventilation rates were adjusted until the median error was minimized on each of the groups of buildings. The results of this calibration are shown in the section 2.3 of the Annex I.

However, this calibration process can be done using an optimization algorithm that finds a compromise between minimizing the squared mean error of the heat demand and the deviation of the ventilation and infiltration rates with respect to some specific reference values from previous studies [PERE14]. Given the characteristics of the model and the way the ventilation rate is computed (see eqn. (3) of Annex I), the behavior of the model is non-linear. As a result, the algorithm selected to perform the optimization does not use any derivatives. This algorithm is the Hypercube optimization method, and it is based on random search, and converges after a process of shrink and displacement [ABIY15]. Regarding the computational power required to perform this task, it is worth pointing out that each simulation of the Junction district, comprised of 823 buildings, takes about 2 hours (Intel® Core™ i7-4790 CPU at 3.60GHz and 16.0 GB of RAM). Since 24 parameters have to be optimized, it is estimated that around 4000-5000 evaluations would be required to achieve a reasonably good accuracy, about 1 year of computational time.

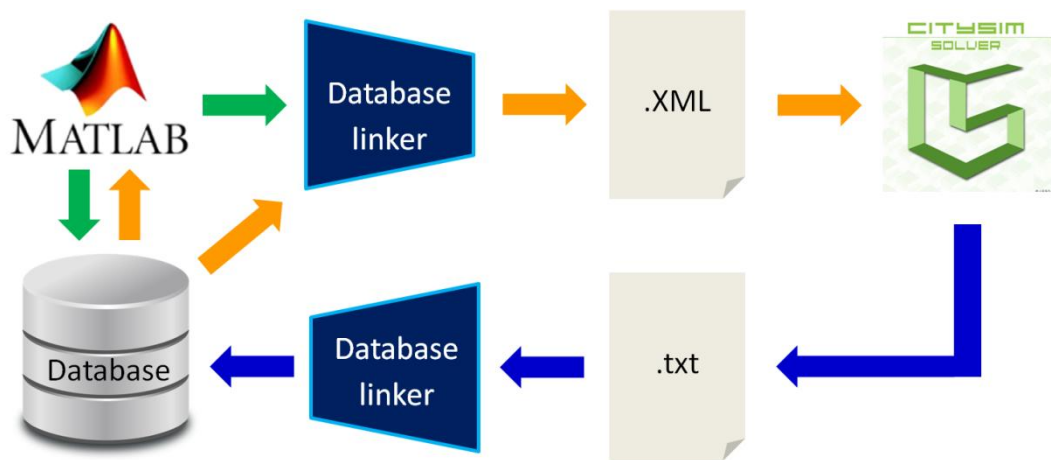


Figure 8: Optimization framework for the calibration process using the Hyper-cube optimization method programmed in MATLAB.

This algorithm was implemented using MATLAB. The program provides the database and the database linker java program with the input parameters that are used to generate the model. The XML file is generated and the buildings simulated in CitySim. Then, another java program puts the results back into the database, and the MATLAB program reads the results from the simulation. The algorithm computes the square mean errors, and the deviation of the input parameters from their respective reasonable reference values. The algorithm assigns one weight to each of these measures of the error, and computes the objective function. The new input parameters for the model are computed and some of them are modified in the database, while others are changed in the database linker java program. This process is repeated iteratively until the objective function converges to an optimal point. One of the advantages of this method is that, if N is big enough, it will likely converge to the global optimum even if there are many local optima.

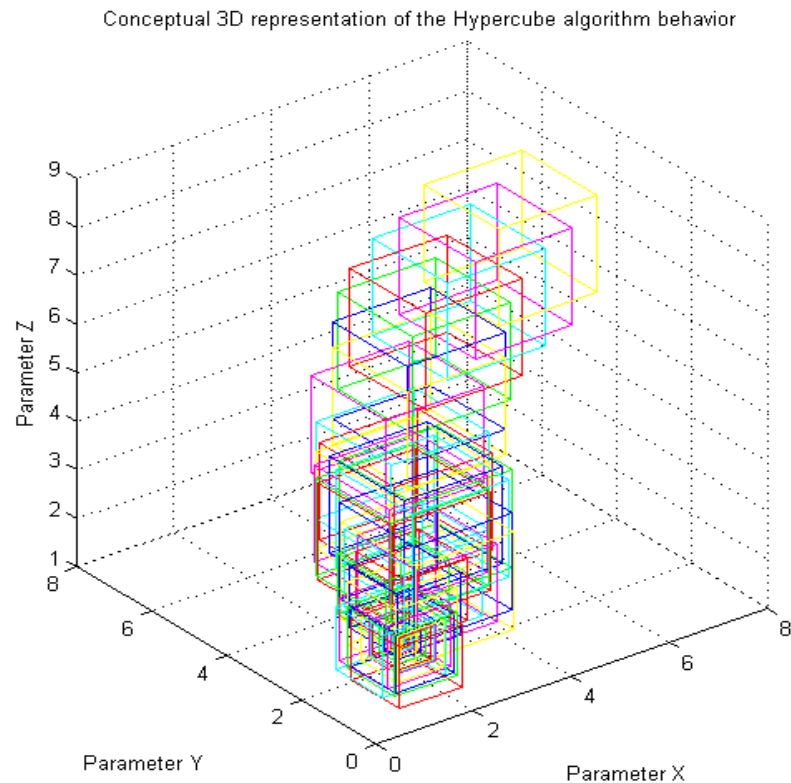


Figure 9: Conceptual 3D representation of the Hypercube optimization algorithm.

Figure 9 illustrates the shrink and displacement process of the hyper-cube. It is initialized at a feasible point in the multi-dimensional space and  $N$  evaluations of the objective function are done inside the hyper-cube. The points to do these evaluations are randomly chosen following a uniform distribution. Then, the center of the hypercube moves towards the point where the cost function was the lowest, and this process is repeated  $M$  times. When the displacement of the hypercube is lower than a threshold, the whole hyper-cube shrinks and becomes smaller. Therefore, the next  $N$  random evaluations will be done in a more limited space. This combination of shrinks and displacements makes this method quite efficient and suitable for the optimization of big dimensional problems. The full MATLAB code of this algorithm is in the Annex III.





## 3 Results and discussion

Once the database is complete and the model calibrated, CitySim allows integrating different kinds of energy systems such as solar PV, solar heaters, boilers, and heat pumps. The electricity consumption and generation of every building can be measured for further analysis. The scope of this section is to show what happens to the energy demand on different refurbishment scenarios, and when PV panels are used to fully exploit the solar potential of the roofs. In addition, the energy signatures of the buildings are analyzed to know more about the potential of certain parameters such as the ambient temperature and the global horizontal irradiance as regressors of the heat demand.

### 3.1 Solar PV integration

The integration of the solar PV on the roofs is explained in section 3 of the Annex I. From the analysis it is possible to observe how we can achieve a net-zero district by refurbishing the buildings properly and taking advantage of the full solar potential of the roofs. However, while some buildings generate more electricity than they need during the daytime, others depend on the grid to satisfy their electricity needs. A well-developed electricity network can help in managing this problem properly. However, some of this energy that is generated during the day must be stored for the hours in which the sun is less intense. Using heat pumps to store heat in heat storage tanks could be a simple option, with the drawback that the heat tanks would require some space available inside the buildings or near them, with the corresponding opportunity cost of allocating that floor area for that purpose. Fuel cells or electric vehicles used as batteries are some other examples of technologies that can help in managing the energy without the need for allocating too much space.

### 3.2 Energy signature of the buildings on an hourly scale

The use of regression models can be useful to predict the energy demand of the buildings based on climatic data and without even running a simulation. Hence, these models can help in predicting the heat demand, managing the heat supply more efficiently, and sizing the equipment based on these estimations. The global horizontal irradiance and the average hourly temperature can be measured in an urban scale and used as regressors. However, the heat demand of highly insulated buildings has a high degree of unpredictability that makes the regression more difficult to perform. This section intends to analyze the goodness of fit of these models considering three different refurbishment scenarios.

These regression methods can be used to size the heating equipment of the buildings [CATA12]. For this purpose, the buildings are typically analyzed without considering the internal gains of the occupants. Therefore, the models whose results are explained in this section take into consideration several refurbishment scenarios, and they do not include the impact of the occupants' behavior for the reasons previously mentioned. Figure 10 shows the distribution of the heat demand as a function of the temperature difference and the unobstructed global horizontal irradiance for two different buildings. As the level of refurbishment increases, the correlation between the temperature difference and the heat demand ( $Q_h$ ) decreases. As expected, the U-value of the buildings also decreases after the building is refurbished. This decrease of the U-value can be seen in the decrease of the slope of the cloud of observations.

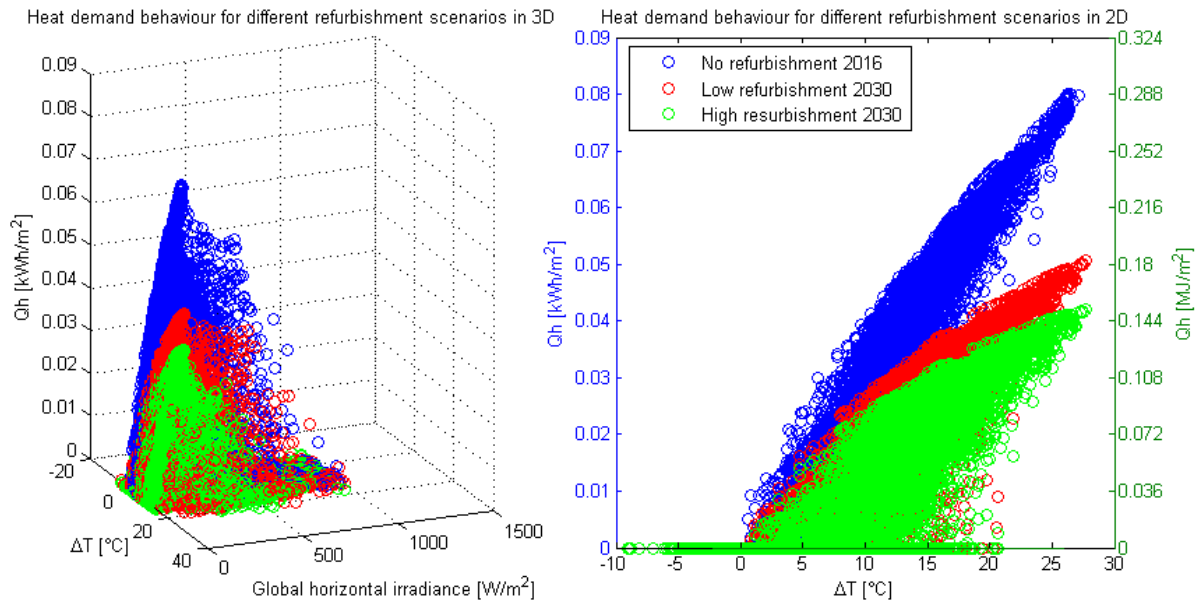


Figure 10: Heat demand as a function of  $\Delta T$  and the global horizontal irradiance for different refurbishment scenarios on an hourly basis.  $R_{2adjusted} = [R_{22016}, R_{2low}, R_{2high}] = [0.8928, 0.7848, 0.7346]$

There is a correlation between the temperature difference  $\Delta T$  ( $T_{in}-T_{out}$ ) and the heat demand of the buildings, which weakens as the degree of refurbishment increases. As a result, the cloud of observations becomes wider and it is more difficult to regress accurately the heat demand using  $\Delta T$ . In Figure 10 the red cloud represents the first level of refurbishment, in which only the windows were replaced. The second level of refurbishment, represented by the green cloud, enhances these aforementioned effects. In this last case, for the building represented by Figure 10,  $\Delta T$  is not a good regressor to estimate  $Q_h$ .

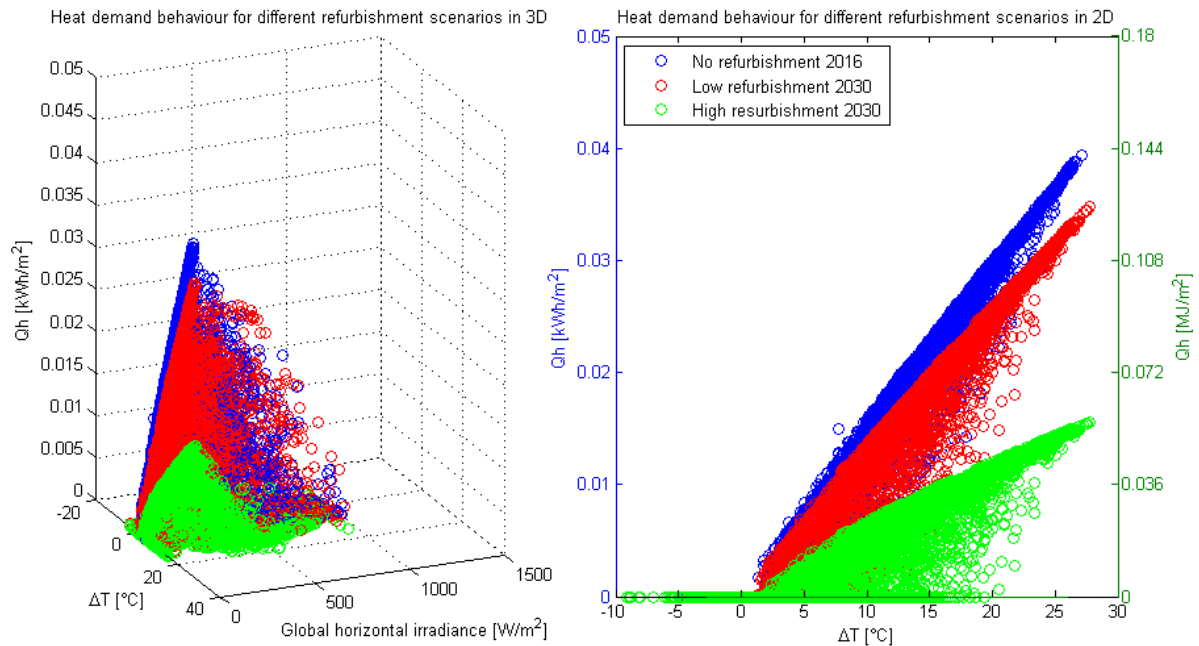


Figure 11: Heat demand as a function of  $\Delta T$  and the global horizontal irradiance for different refurbishment scenarios on an hourly basis.  $R^2_{adjusted} = [R^2_{2016}, R^2_{low}, R^2_{high}] = [0.9599, 0.9309, 0.8176]$

Figure 11 illustrates the heat demand of a different building, which has a high correlation between  $\Delta T$  and  $Q_h$ . For the high refurbishment scenario,  $Q_h$  can be estimated with reasonable accuracy for values of  $\Delta T$  greater than 20°C. However, as  $\Delta T$  decreases so the cloud dispersion does, increasing the margin of error of any possible regression.

In both Figure 10 and Figure 11 we can observe that there is a small correlation between  $Q_h$  and the global horizontal irradiance. As the irradiance increases, the range of  $Q_h$  becomes narrower, and  $Q_h$  is therefore slightly easier to determine taking into account the irradiance as a regressor. However, this correlation between  $Q_h$  and the horizontal irradiance is not the same for the buildings whose windows are unobstructed than for those that are shadowed. It is also important to take into account that the intensity of the irradiance follows a sinusoidal behavior, which can be non-linear due to the shadowing effects. Therefore, the global horizontal irradiance does not seem to be a good regressor, but it is worth analyzing the adjusted  $R^2$  of all the buildings and comparing them for different variables used as regressors ( $\Delta T$  and the horizontal irradiance), as Figure 12 depicts.

Figure 12 shows the distribution of the adjusted  $R^2$  of the buildings when both  $\Delta T$  and the global horizontal irradiance are considered as regressors, and when only the  $\Delta T$  is considered. The graph shows that both distributions are very similar. Although  $R^2$  is slightly higher when both  $\Delta T$  and the global horizontal irradiance are taken into account, the increased complexity of using two regressors might not be worth for such a small improvement of the goodness of fit.

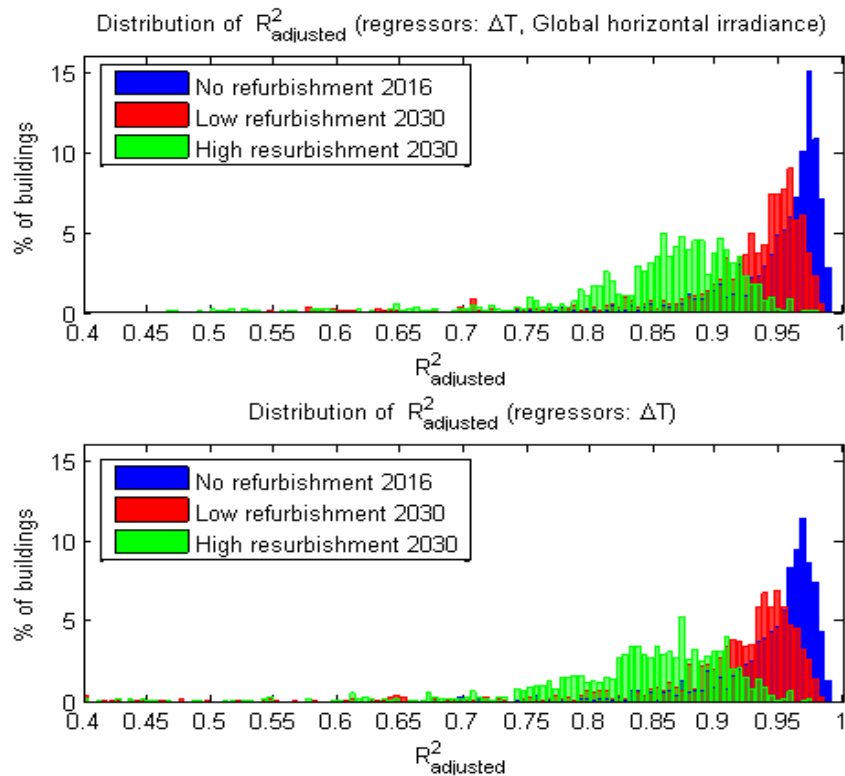


Figure 12: Distribution of the adjusted R2 of the heat demand comparing various regressors on an hourly basis.

As mentioned before, the occupants were removed from the model when analyzing these correlations between the  $Q_h$ ,  $\Delta T$ , and the global horizontal irradiance. The reason for this was to study the buildings in a more ideal way, without the stochastic internal gains due to the occupants or the movements of the blinds. However, it is also interesting to analyze the model in terms of their occupants' behavior. Figure 13 compares the adjusted  $R^2$  between two models, one of them without occupants and the other with them. The model with occupants includes a limit of  $200\text{W}/\text{m}^2$  to  $400\text{W}/\text{m}^2$  (depending on the type of building) for the irradiance that can enter through the windows. In the model without the occupants this limit is  $1360\text{W}/\text{m}^2$  in order to consider that the blinds are always open. As Figure 13 shows, the regression in the model that does not consider the occupants' behavior is more accurate than in the model that does include the occupants. This is because the stochastic behavior of the occupants adds some more uncertainty to the model. The decrease of the goodness of fit is particularly more significant in the high refurbishment scenario because highly insulated buildings are more sensible to the internal gains, and their  $Q_h$  is therefore more difficult to predict.



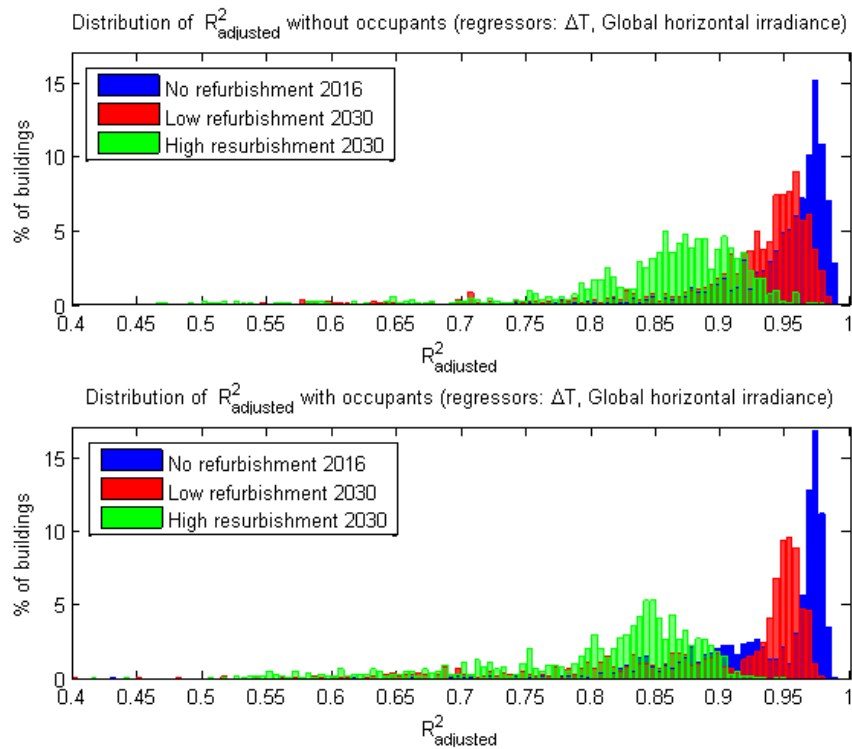


Figure 13: Distribution of the adjusted R2 of the heat demand with  $\Delta T$  and the global horizontal irradiance as regressors. The effect of the occupants on these distributions is analyzed and compared between both graphs (the upper graph does not include the occupants, while graph at the bottom does).

Figure 14 shows the energy signature of two different buildings that include the internal gains due to the occupants' behavior. The first of them has a higher correlation between  $Q_h$  and  $\Delta T$  than the second one, without any very significant distortion caused by the occupants. This building has a big wall surface that exchanges heat with the ambient and, therefore, this building is much more sensible to the climatic conditions than it is to the behavior of the occupants. The graph below, also in Figure 14, depicts a building with a different behavior of  $Q_h$  with respect to  $T_a$ . The 3D plot shows a weak correlation between the global horizontal irradiance and  $Q_h$ , and no variation of  $Q_h$  for irradiance values greater than  $400\text{W/m}^2$ . This is due to the limitation of the solar gains imposed by the automated blinds, which were taken into account for this model with occupants but not for the models previously mentioned that had no occupants. The MATLAB code written to obtain all these results is included in the Annex II.

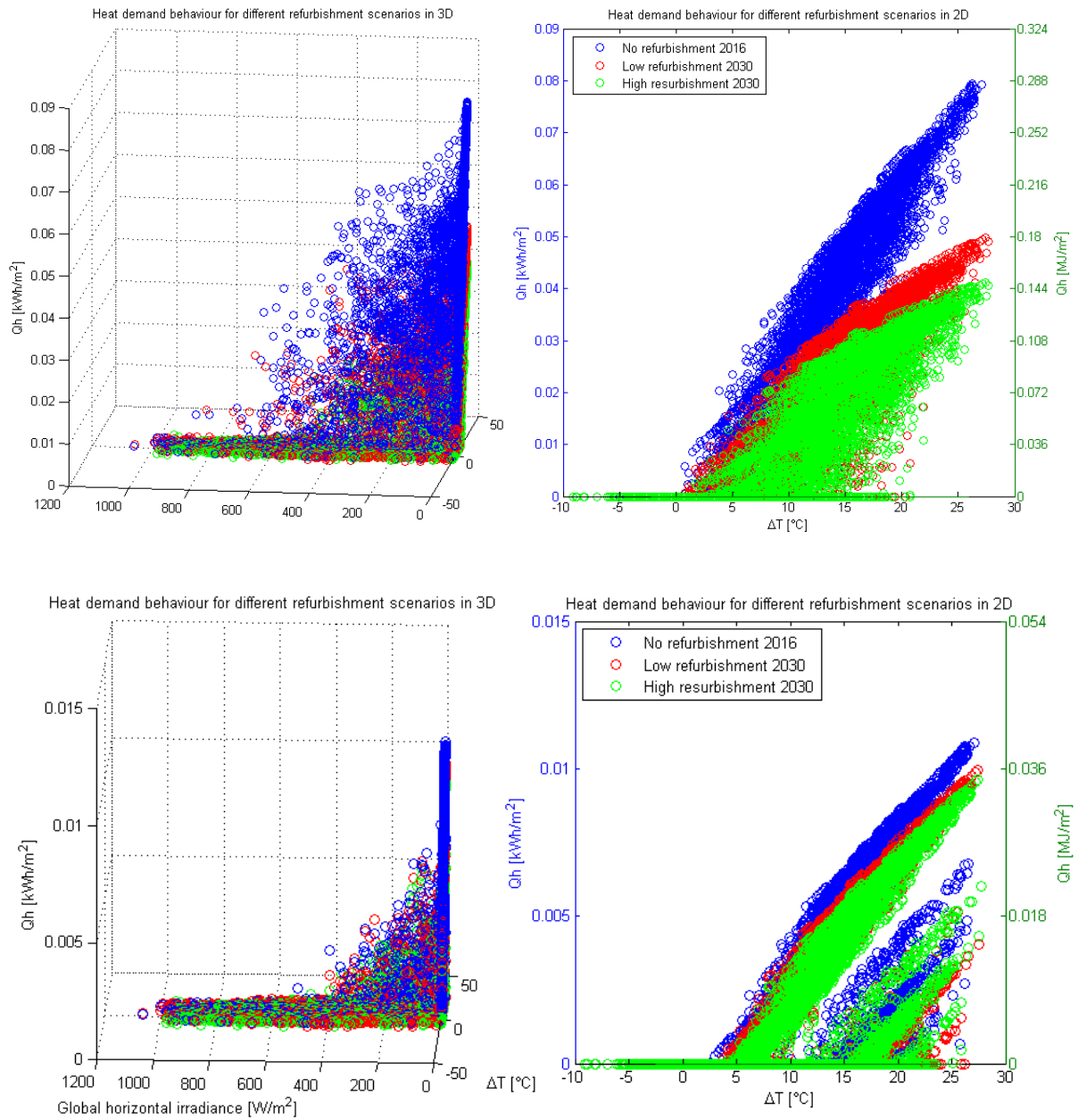


Figure 14: Heat demand as a function of  $\Delta T$  and the global horizontal irradiance for different refurbishment scenarios on an hourly basis and including the occupants.  $R2_{adjusted} = [R2_{2016}, R2_{low}, R2_{high}] = [0.8802, 0.7637, 0.7125]$  (figures above);  $R2_{adjusted} = [R2_{2016}, R2_{low}, R2_{high}] = [0.3764, 0.3333, 0.3775]$  (figures at the bottom).

The 2D plot of the second graph in Figure 14 shows a shift of the energy signature for many of the observations, forming several parallel bands under the main distribution. These bands are the result of the stochastic behavior of the internal gains of this building as a consequence of its occupancy profiles during certain hours of the day. This is a commercial building, which has an intense occupancy rate,

and during the weekends this occupancy was set to zero, which can explain the existence of more of these bands under the main dispersion.

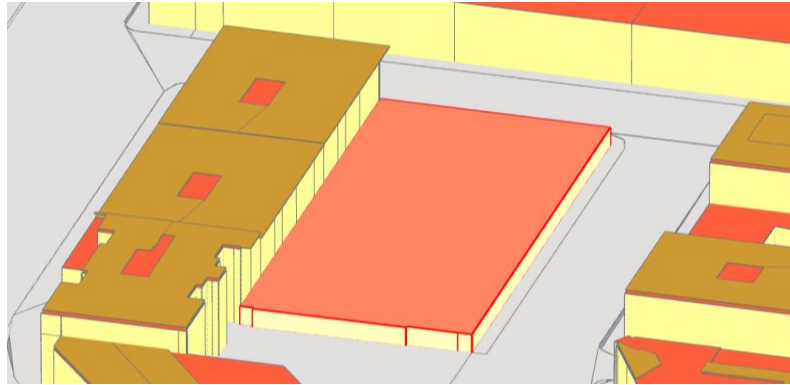


Figure 15: Building whose thermal behavior is depicted by the graph at the bottom of Figure 14.

Figure 15 shows the aforementioned commercial building whose energy signature is depicted by the graph at the bottom of Figure 14. Its wall surfaces are very small in comparison to its volume, and therefore, the impact of  $\Delta T$  on  $Q_h$  is lower than it is in other buildings. In addition, the building has a high occupancy rate, which causes the correlation between  $Q_h$  and  $\Delta T$  to decrease exactly as Figure 14 suggests.

### 3.3 Energy signature of the buildings on a daily scale

It is important to remark that the results shown in this section are expressed on a daily basis, which means that each observation is the total heat demand of a given day. The heat demand is expressed in  $\text{kWh/m}^2$  in order to compare the graphs of buildings of different sizes. This daily analysis of the results ignores the hourly variations of the heat demand within each day. As a consequence, the analysis presented in this section has strong limitations in terms of real applicability to dimension and control the heating equipment. For these purposes, it is better to take into account the previous analysis on an hourly basis, and find an optimal dimension for both the heat tank and the heating equipment without over-sizing any of them unnecessarily. However, the daily analysis of the heat demand can have some applications, for example, to find an upper bound of the heat tank's volume.

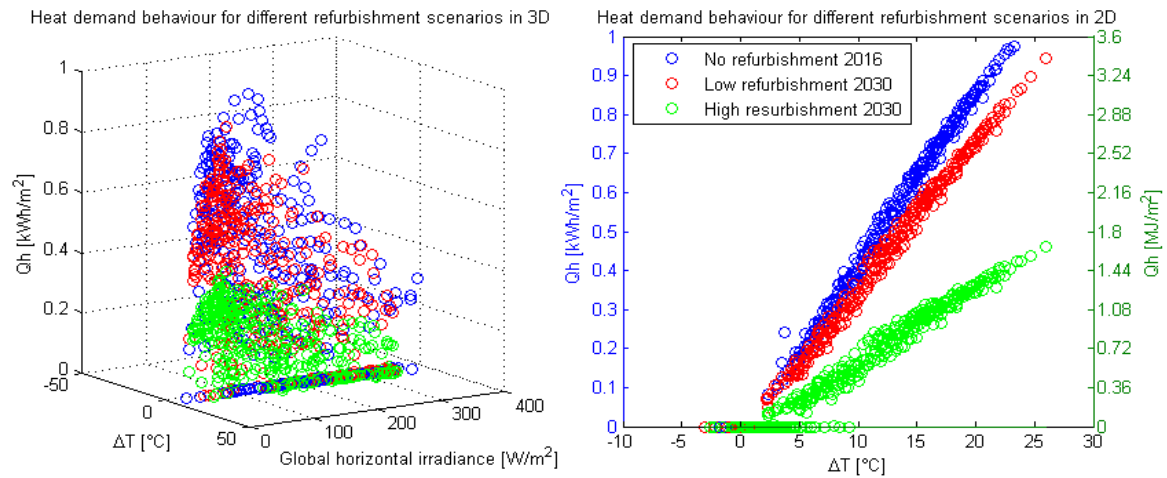


Figure 16: Heat demand as a function of  $\Delta T$  and the global horizontal irradiance for different refurbishment scenarios on a daily basis.  $R2_{adjusted} = [R2_{2016}, R2_{low}, R2_{high}] = [0.9911, 0.9890, 0.9570]$

Figure 16 depicts the energy signature of a building on a daily basis during a year. The correlation between  $Q_h$  and  $\Delta T$  is higher than in the hourly analysis because the effect of the large variations of  $\Delta T$  between day and night is ignored. However, the daily heat analysis can still provide some useful information. In order to size the heat tanks in such a way that they absorb the hourly variations by storing heat during the day, and releasing it at night (when the heating demand is typically higher). The heat demand of the day of maximal heat demand can help to estimate a very conservative upper bound of the heat tank's volume. Following a similar reasoning, but for the hourly analysis, knowing  $Q_h$  during the hour of maximal heat demand is useful to set an upper bound for the power of the heating equipment. Since heat tanks can also be used to store energy in the hot seasons and release it during the cold seasons, a daily analysis of the heat demand in which the daily observations are classified in clusters that represent the different seasons could help to size the heat tanks and the heating equipment as well.

Figure 17 shows the distribution of the adjusted  $R^2$  for the different refurbishment scenarios in the daily analysis. In the graph above, both  $\Delta T$  and the global horizontal irradiance were considered, while in the graph below, only  $\Delta T$  was taken into account. In this case, the difference between using two regressors instead of just one is slightly greater than in the hourly analysis.

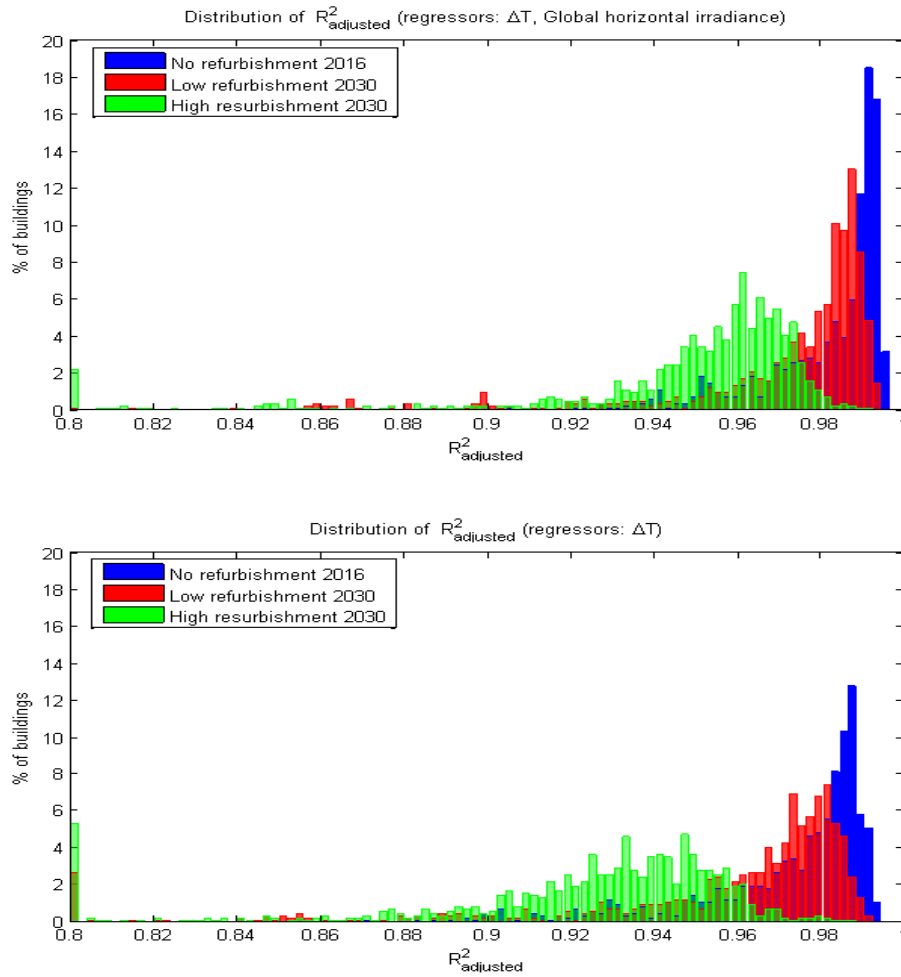


Figure 17: Distribution of the adjusted R2 of the heat demand for different refurbishment scenarios and using different combinations of regressors on a daily basis.

As previously mentioned, the daily analysis can provide some useful information to be used in setting an upper bound to the volume of the heat tanks that would be enough to absorb the  $Q_h$  variations between the cold and the hot periods of the day, and assuring maximal comfort. Eqn. 4 was used to calculate the volume of the heat tanks of each building by using the peak  $Q_h$  of the year.

$$V_{tank} = \frac{Q_{h_{maxDay}}}{\rho_w \cdot C_{p_w} \cdot \Delta T_{tank-building}} \quad (4)$$

Figure 18 shows the results of applying this equation in different refurbishment scenarios to all the 823 buildings of the Junction district. The cumulative volumes are represented by the corresponding curves, which are helpful to provide a better understanding of the magnitude of the storage capacity if a centralized system were built.

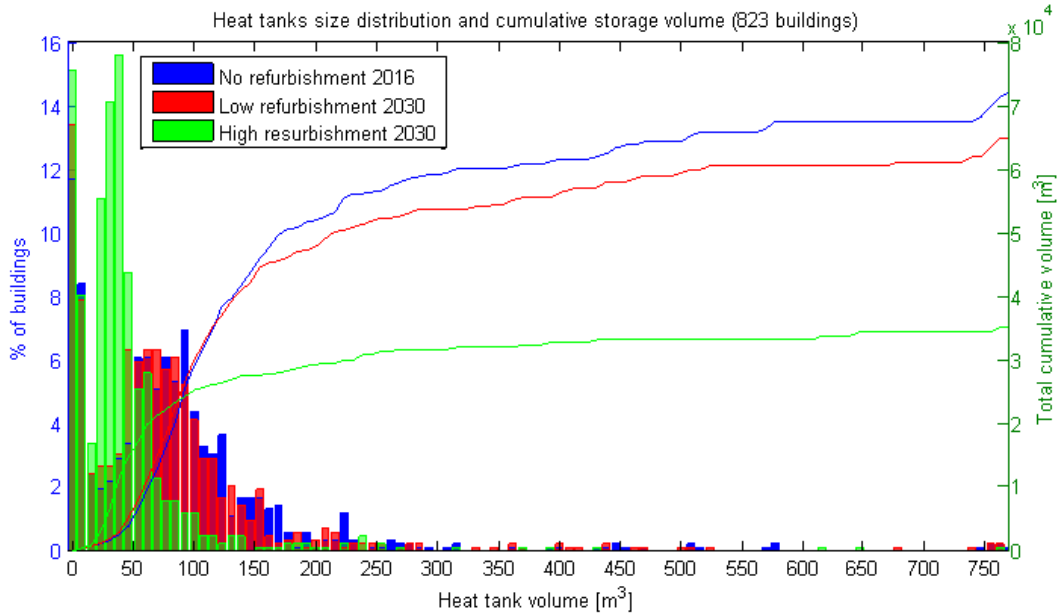


Figure 18: Distribution of the heat tank volume of the buildings for different refurbishment scenarios and assuming that the tank would be able to store all the heat that would be consumed in the day of maximal heat demand of its associated building.

From the total cumulative volume it is possible to calculate that the average size of the heat tanks in the low refurbishment scenario is  $79\text{m}^3$ , while in the high refurbishment scenario the average volume per building is  $42.5\text{m}^3$ . As mentioned before, these values should just be used as a reference of the upper bounds of the heat tank volumes if we wanted to absorb the intra-day variations of  $Q_h$ . However, since the heat tanks are only charged during the hottest hours of the day, it would be reasonable to reduce these volumes by about 30%-40% and still obtain a realistic upper bound of the heat tanks' volumes. When an air to water heat pump is used, the heat tank helps to store the excess heat it produces when its COP is higher due to the high ambient temperature, which is  $T_{\text{evap}}$  in eqn. (5).

$$COP = \eta \cdot \frac{T_{\text{cond}}}{T_{\text{cond}} - T_{\text{evap}}} \quad (5)$$

The use of large heat tanks has some drawbacks. First of all, due of the need for enough space available to place the tanks and the opportunity cost that this involves for the owners of the building. For instance, the price of the floor surface of an apartment in Geneva is around 8,000-10,000CHF [NUMB16]. Furthermore, if the tanks were designed to satisfy the heat demand during the day of maximal demand, they would not be used at full capacity most of the time. Heat losses would be another issue, although in this case, larger heat tanks would actually help to reduce them. A large heat tank can store the same amount of heat than a small one but at a lower temperature, and therefore, reducing its thermal losses. In addition, the larger the tank, the smaller its external surface to volume rate is, and the lower its losses. However, the higher investment cost, and the inconvenience for the user of allocating the required space for the heat tank are more powerful reasons to not oversize the heat tanks.

For these reasons, it is important to analyze the heat demand on an hourly basis and find a good compromise among the size of the heat tanks, the heating equipment, and also the heat exchangers in the case of a district heating network, while the daily analysis can simply be used to estimate the upper



bound of the heat tanks' volumes before starting the optimization process. Therefore, the next section is intended to analyze the thermal comfort as a function of the volume of the heat tanks and the power of the heating equipment (heat pumps in this case). Finally, this comfort is quantified in economic terms, taking into account the investment and the variable costs.

### 3.4 Heat pump and heat storage tank dimensioning

Adding a heat tank to store energy by heating up water can help in reducing the rated power of the heating device to be used. In this case, an air to water heat pump can be combined with this type of tank, which the heat pump will heat up during the periods of excess power capacity (typically during the day). When the heat pump lacks the power to fully supply the demand, the heat tank will help in meeting the energy target. As a consequence, it is possible to install heat pumps with a lower rated power without decreasing the comfort achieved. The main drawbacks of this method are the increased energy required due to the losses of the tank, the amount of space required to install the water heat tank (in Geneva the price of each square meter is about CHF 8,000-10,000), and the additional investment cost of the heat tank (although the investment cost of the heat pump is reduced). Therefore, it is worth analyzing the thermal comfort as a function of the annualized costs of the heat pump and the heat tank, which includes both CAPEX and OPEX.

The thermal comfort can be measured with the PMV and PPD equations (Predicted Mean Vote and Predicted Percentage of Dissatisfied). Both equations are linked to each other by eqn. (7) and for the purpose of this study only the Predicted Percentage of Satisfied (100%-PPD) was analyzed as a function of the size and cost of the heat tank and the heat pump. There are many factors have a direct effect on the PPD, the most relevant ones and their estimated values are provided below.

$$PMV = -0.04 + (0.303 \cdot e^{(-0.036 \cdot M)} + 0.028) \cdot ((M - V) - 3.05 \cdot 10^{-3} \cdot (5733 - 6.99 \cdot (M - V) - Pa) - 0.4 \cdot ((M - V) - 58.15) - 1.7 \cdot 10^{-5} \cdot M \cdot (5867 - Pa) - 0.0014 \cdot M \cdot (34 - Ta) - 3.96 \cdot 10^{-8} \cdot fcl \cdot ((tcl + 273)^4 - (Trm + 273)^4) - fcl \cdot hcl \cdot (tcl - Ta)) \quad (6)$$

$$PPD = 95 \cdot e^{(-0.03353 \cdot PMV^4 - 0.2179 \cdot PMV^2)} \quad (7)$$

$$hcl = 12.1 \cdot \sqrt{Va}$$

$$fcl = 1 + 1.290 \cdot Icl \quad \text{if } Icl \leq 0.078$$

$$fc = 1.05 + 0.645 \cdot Icl \quad \text{if } Icl > 0.078$$

$$M = 100 \text{ W/m}^2$$

$$V = 0 \text{ W/m}^2$$

$$Icl = 0.5 \cdot 0.155$$

$$RH = 50\%$$

$$Pa = (10 \cdot RH) \cdot e^{(16.6536 - \frac{4030.183}{Ta + 235})}$$

$$Tg \approx Ta$$

$$Ts \approx Ta$$

$$Trm = Tg + 1.9 \cdot \sqrt{Va} \cdot (Tg - Ts)$$

$$Ta$$

$$Va = 0.1 \text{ m/s}$$

*heat transfer coefficient in W/m<sup>2</sup>K*

*clothing area factor*

*Metabolic rate*

*Effective mechanic power*

*Intrinsic clothing insulation in K·m<sup>2</sup>/W*

*Relative humidity*

*water partial pressure in kPa*

*Wet bulb temperature (approximation)*

*Dry bulb temperature (approximation)*

*Radiant mean temperature*

*Air temperature in °C*

*Air velocity*

The analysis was performed on a single building and the algorithm, which it is included in Annex VI, was implemented in MATLAB. This algorithm sets the power of the heat pump and the volume of the heat tank to a desired value, it runs the simulation with CitySim, reads the results, and computes the PPD as a function of the internal temperature of the building and the rest of the estimated

parameters shown below eqn. (7). All the results are saved and the process is iterated for many different values of volume and rated power. The results from this process are depicted in Figure 19.

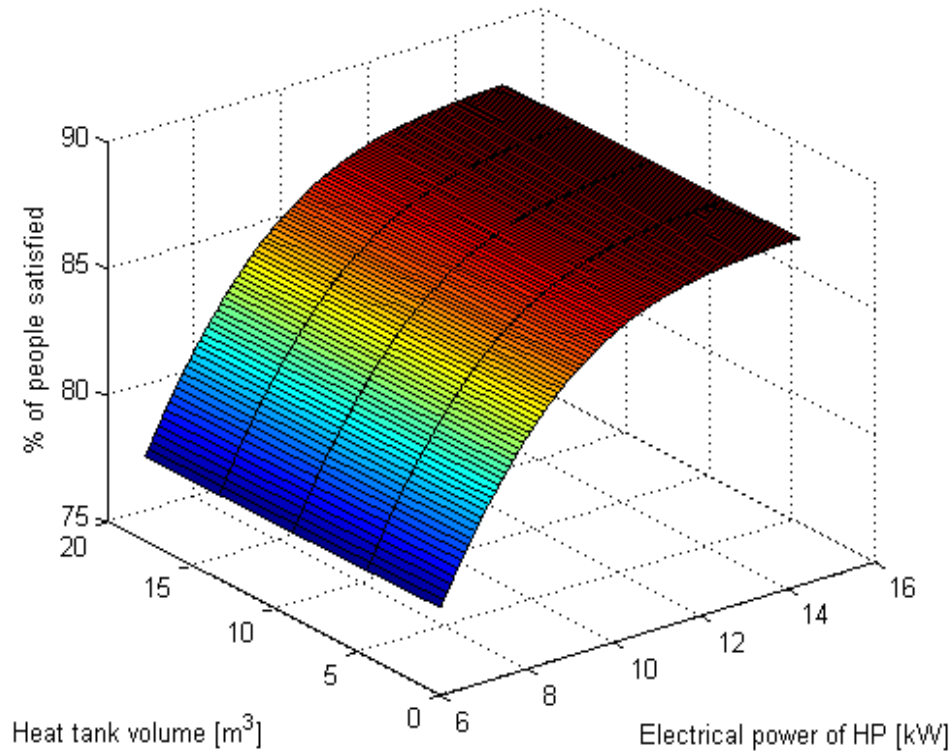


Figure 19: Percentage of people satisfied as a function of the heat tank volume and the HP power.

Surprisingly, the volume of the heat storage tank did not seem to have any major effect neither on the comfort of the occupants nor on the internal temperature of the building. This result was unexpected and, after analyzing the data obtained from the simulation, the problem was found to be in the algorithm used to control the heat pump – heat tank interaction. The simulation tool CitySim uses a simplified control algorithm of the temperature of the heat tank that makes the heat pump supply heat to the tank if, and only if, its temperature is equal to  $T_{min}$  or less. As a consequence, the heat pump heats up the heat tank for a single hour and stops when it is already above  $T_{min}$ . The result is that the heat tank rarely reaches  $T_{max}$  and, therefore, its maximal thermal capacity.



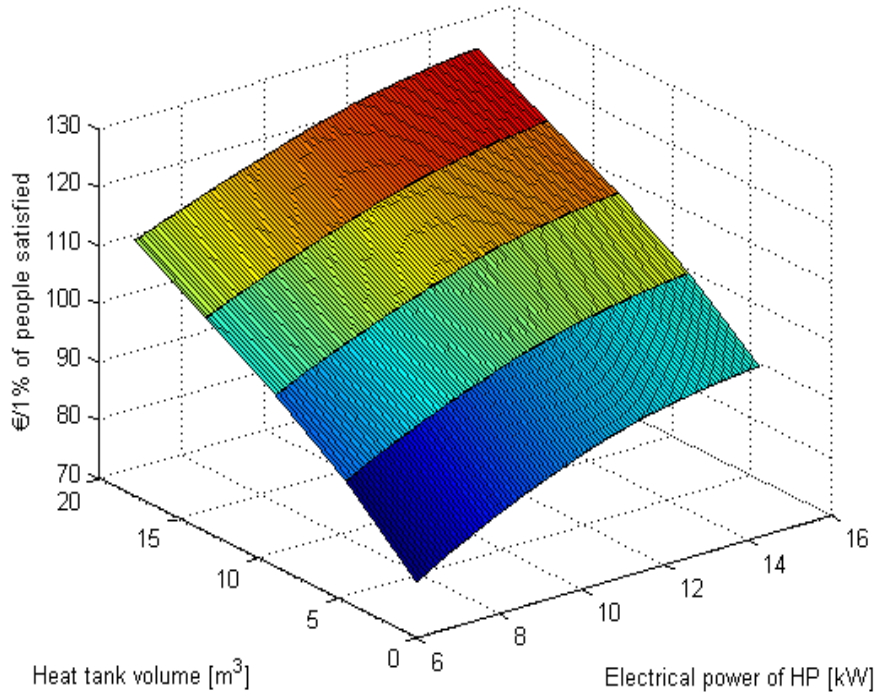


Figure 20: Price of comfort as a function of the heat tank size and the power of the heat pump.

Figure 20 illustrates the annualized cost of the percentage of people satisfied as a function of the heat tank volume and the power of the heat pump. Since the algorithm that controls the temperature of the heat tank does not work well, the cost per unit of comfort (% of people satisfied in this case) only increases as the heat tank becomes larger. However, with a good control system, we could expect Figure 19 and Figure 20 to show that it is possible to achieve a specific level of comfort by reducing the size of the heat pump and increasing the volume of the heat tank. It is worth combining both Figure 19 and Figure 20 in order to first set a desired level of comfort, and then choose the most cost-efficient solution to reach that level of comfort.

The annualized costs were estimated based on the investment and the variable costs of these devices. The investment cost of the heat pump was estimated interpolating (and extrapolating) the investment costs of real heat pumps [HPPG16]. Although there are equations that can be used to calculate the investment costs of industrial equipment, this interpolation method with real data was applied instead due to the non-industrial usage of this machinery and its limited power, which is much lower than the typical rated power of industrial equipment. Eqn. (8) is used to estimate the price of the floor surface to be allocated to place a heat tank of volume  $vol$  in Geneva. Eqn. (9) is used to annualize any investment costs for  $n$ -years and a given interest rate  $i$ .

$$I_{spaceTank} = \pi \sqrt[3]{\left(\frac{vol}{2\pi}\right)^2} \cdot \frac{8000 \cdot CHF/m^2}{1.086 \cdot CHF/€} \quad (8)$$

$$I_{annualized} = I \cdot \frac{i \cdot (1+i)^{years}}{(1+i)^{years} - 1} \quad (9)$$



Eqn. (10) is used to calculate the annualized CAPEX as the sum of all the annualized investment costs. Eqn. (11) calculates the OPEX, or variable annual costs, which in this case have been assumed to be the price of electricity.  $P_e$  is the electrical energy consumed in kWh.

$$CAPEX_{annual} = I_{spaceAnnualized} + I_{tankAnnualized} + I_{hpAnnualized} \quad (10)$$

$$OPEX = P_e \cdot 0.21 \cdot \frac{CHF}{kWh} \cdot \frac{1\text{€}}{1.086CHF} \quad (11)$$

$$Annualized\ cost = CAPEX_{Annual} + OPEX \quad (12)$$

The investment costs of the heat storage tank were estimated based on empirical data as well [SRWT16], while the heat losses were estimated considering an insulated heat tank with 4[Btu/ft<sup>2</sup>h] of losses [ET16]. The heat tank was assumed to be cylindrical and with a height equal to its diameter. This shape was chosen because it minimizes the external surface to volume ratio, and therefore, it minimizes the heat losses. Eqn. (13) shows the mathematical relationship between the volume of the heat tank and its heat losses.

$$Heat\ tank\ losses\ \left[\frac{W}{K}\right] = \frac{4 \frac{BTU}{ft^2 \cdot h} \cdot \left( 2\pi \left(\frac{vol}{2\pi}\right)^{\frac{2}{3}} + \left( 4\pi \left(\frac{vol}{2\pi}\right)^{\frac{2}{3}} \right) \right) m^2 \cdot \frac{0.29307 \frac{W}{BTU/h}}{\left(0.3048 \frac{m}{ft}\right)^2}}{(33 - T_{building}) K} \quad (13)$$

In addition, and given that the heat tanks can be quite large, the cost of the floor surface required to install the tank was taken into account. An estimated price of 8,000 CHF/m<sup>2</sup> was considered. However, when annualizing this cost, a very long period of amortization time was used (considering the investment cost of the land as a perpetual annuity). The reason is that the land that is used to install the tank does not perish as a heat pump, an apartment or a tank does.

The investment costs were annualized using eqn. (9) in order to compute the annualized CAPEX. The OPEX was estimated with the annual cost of the electrical consumption, a price of 0.21CHF/kWh was applied [ALP16], and the maintenance costs of the equipment were neglected. Eqn. (10) shows the total annual cost as the sum of the annualized CAPEX and OPEX. However, in a centralized heating system, the heat pump could be considered as an industrial device, and its investment cost could be calculated as the sum of investment costs of its compressor and heat exchangers. Eqn. (14) would be used to calculate the bare module cost as the sum of the direct and indirect costs [BEND15]. The direct costs calculated in eqn. (15) include the costs of the purchase, the materials, and the installation, while the indirect costs of eqn. (16) include the freight, the construction overhead and the contractor engineering expenses. All other costs that contribute to the bare module cost can be considered to be proportional to the purchase cost by a bare-module factor  $F_{BM}$ , which is calculated using eqn. (19).

$$C_{BM} = C_d + C_i \approx F_{BM} \cdot C_p \quad (14)$$

$$C_d = F_{BM} \cdot C_p + C_m + C_{ins} \quad (15)$$



$$C_i = C_f + C_o + C_e \quad (16)$$

The purchase cost of the equipment is computed with the empiric formulas in eqn. (17-18), which are derived from a cost database of existing equipment. The constants  $k_i$  are empirical and based on the equipment costs available in the database. Eqn. 17 is used to compute the costs of the heat exchangers based on their heat exchange area  $A$ . Eqn. 18 calculates the cost of the compressor based on its power  $P$ .  $I_t$  is an industrial equipment cost index for the current year, and  $I_{t,ref}$  is the same cost index for the reference year from which the empirical constants are taken.

$$C_p^{HEX} = \frac{I_t}{I_{t,ref}} \cdot 10^{k_1 + k_2 \log A + k_3 (\log A)^2} \quad (17)$$

$$C_p^{COMP} = \frac{I_t}{I_{t,ref}} \cdot 10^{k_1 + k_2 \cdot \log P + k_3 \cdot (\log P)^2} \quad (18)$$

The purchase cost is then multiplied by the bare-module factor, which takes into account the influence of the operating pressure and the materials of construction, eqn. (19). The constants  $B_i$  are calculated on the base of existing equipment cost databases,  $F_m$  is the material factor and  $F_p$  is calculated with eqn. (20) and takes into account the effect of the operating pressure change on the bare-module cost. The  $F_p$  is important to determine the cost of the heat pump compressor.

$$F_{BM} = B_1 + B_2 \cdot F_m \cdot F_p \quad (19)$$

$$F_p = 10^{c_1 + c_2 \cdot \log P + c_3 \cdot (\log P)^2} \quad (20)$$

Finally, the termed total module cost  $C_{TM}$  includes the effect of any contingencies and fees associated to the bare module cost. A typical value of  $\alpha$  is 0.18.

$$C_{TM} = (1 + \alpha)C_{BM} \quad (21)$$

It is necessary to improve the algorithm that controls the heat tank's temperature before properly analyzing the advantages of using a heat tank of a certain volume and reaching meaningful conclusions. Therefore, in the next section an improved control algorithm is proposed for future implementation.

### 3.5 An improved control algorithm for the heat pump – heat tank system

The main drawback from storing as much energy in the heat tank as possible is that we can be heating it up more than we need to satisfy the next shortage cycle. As a result, the temperature of the heat storage tank will always be above its minimal value, and therefore, the thermal losses will be higher than necessary. On the other hand, if the heat tank is not heated up to the temperature that is necessary to satisfy the energy demand during the next period of shortage, the temperature of the building will not reach its target level and the thermal comfort will decrease. Therefore, it is important to find a compromise between thermal comfort and energy losses of the tank. In the ideal scenario, the heat tank will store as much heat as it is needed to fully satisfy the heat demand of the building during the next shortage (probably during the night), and at an optimal cost.



A Model Predictive Control (MPC) can help this system in achieving a good performance [JIN14]. However, it is also possible to achieve a better performance than the current one by implementing a simple PID control. Although it might not have the potential of achieving as optimal results as the MPC, it is simpler and easier to program. Therefore, as a first improvement for the control system of the heat pump and the heat tank in CitySim, a PID control is proposed and explained in this section.

This proposed algorithm in eqn. (22) calculates the error of the system as the difference between the shortage of heat in the last 24 hours in the building, and the amount of heat that is currently stored in the heat tank available to be used. Hence, a positive error means that the combination of the heat pump and the storage tank are not capable of fully satisfying the heat demand of the building. Therefore, the controller increases the amount of energy that is stored in the heat tank (assuming that the heat pump has enough excess power to do so). Since, at the steady state, the heat shortage of the building would ideally be zero, it is important to specify that  $Q_{ref}$  at time  $j$  is equal to  $Q_{ref}$  at the same hour of the day before ( $j-24$ ) plus the additional command of the PID at time  $j$ . This is a way of taking into account the cyclical behavior of the system, and of preventing the heat tank from remaining empty if in the last 24 hours there has not been any energy shortage in the building. This control system will probably lead to a small surplus of stored heat at some point due to the over-shooting; which will be removed by the controller itself after a few iterations by applying a negative command.

$$\text{Heat to be supplied to the heat tank: } Q_{ref\ j} = Q_{ref\ j-24} + Q_{Pj} + Q_{Ij} + Q_{Dj} \quad (22)$$

$$e_j = (m \cdot Cp)_{tank} \cdot (T_{tank\ j} - T_{min}) - (m \cdot Cp)_{building} \sum_{i=j-23}^j (T_{ref} - T_i) \quad \text{where } T_{ref} = 21^\circ\text{C}$$

$$Q_{Pj} = K_P \cdot e_j \quad \text{Proportional action}$$

$$\begin{cases} Q_{Ij} = Q_{Ij-1} + \tau_s \cdot K_I \cdot e_j & \text{if } P_{HeatPump} < P_{max} \\ Q_{Ij} = 0 & \text{if } P_{HeatPump} = P_{max} \end{cases} \quad \text{Integral action}$$

$$Q_{Dj} = \frac{K_D}{\tau_s} (e_j - e_{j-1}) \quad \text{with a saturation value of } \pm Q_{D\ \text{saturation}} \quad \text{Differential action}$$

where  $\tau_s$  is the sample period (1 hour), and  $K_P$ ,  $K_I$ , and  $K_D$  are the gains of the control

The gains  $K_P$ ,  $K_I$ , and  $K_D$  should be calibrated to maximize the speed of the controller without being too under damped, otherwise the temperature of the heat tank would increase far beyond of what is needed due to the over-shooting, and would not decrease until the next cycle. This would increase the heat losses of the heat tank more than necessary. PID controllers, as the MPCs, have been used for these types of applications in many studies [ISLA13] [WASE13].



## 4 Conclusion

The rising urban population, combined with the increasingly demanding CO<sub>2</sub> target levels of CO<sub>2</sub> emissions, is a challenging goal that many governmental and municipal authorities are facing today. Building simulation at the urban scale can be an effective approach to provide cities with useful data about the impact of specific policies on, for instance, building refurbishment or renewable energy technologies integration. This allows adopting cost-efficient measures whose impact is maximized, by for example, targeting more easily sets of buildings that have a high potential to reduce their energy demand. However, availability of data and need for high computational power are some important technical limitations, especially when performing optimization processes on these types of models. From the social and business perspective, companies that hold most of the data that can be useful to calibrate these models, as electricity consumption data, do not have any economic incentive to share it. In addition, these data are very sensitive and must usually be aggregated before being shared.

For the development of a model to be complete, it is necessary to estimate many of the parameters of the buildings. To perform this task, it is crucial to properly classify the buildings in a meaningful way that allows using statistical data to make good estimates. For instance, the age of construction and refurbishment, combined with the location of the building, are useful information to estimate the construction materials, the insulation, and the infiltration rates. The occupancy type of the building can help to predict the occupancy and electrical consumption profiles, and also the ventilation rates. The thermal inertia, which is related to the number of floors, has a significant effect on the heat losses, and therefore, on the heat demand. The results show that for certain buildings with high solar gains, the thermal inertia played an important role in decreasing the heat losses with respect to the same buildings modeled with a lower thermal inertia. As a conclusion, in order to optimize the computational time of the simulations, buildings whose solar gains are high should be modeled with as many thermal zones as floors. On the other hand, buildings that are highly shadowed, and whose internal gains due to the occupants are not very high, can be modeled with a single thermal zone without any significant difference with respect to the same building modeled with several thermal zones.

The main challenge of the calibration process is to adjust some of the building attributes to make the behavior of the model as real as possible without assigning unrealistic values to those attributes. This problem can be addressed by defining an objective function that takes into account the squared mean error of the heat demands, and also the deviations of the values given to the input parameters with respect to their reference values. The optimization algorithm used to calibrate this model should not be based on derivatives due to the non-linear characteristics of the model. The hyper-cube optimization method is an effective algorithm to perform this task, and the main limitation of this optimization process was the required computational time.

The buildings' energy signatures are widely used to estimate the heat demands without performing any simulation of the building energy system itself. Although this approach can have an acceptable accuracy for certain buildings, especially during the winter season, once these buildings are refurbished and are less sensitive to the ambient temperature and more to the internal gains this accuracy dramatically decreases, and the ambient temperature can no longer be used as a regressor to predict the heat demand accurately. Buildings with a high external wall surface to volume ratio are more sensitive to the ambient temperature and their heating demand easier to predict, especially if they are not very exposed to sunlight and have low occupancy rates. The global horizontal irradiance, by itself, does not seem to be a good predictor of the heat demand, although it can slightly increase the goodness of fit of a linear regression model when it is combined with the ambient temperature. Due to



the location of the buildings and the geometrical particularities of the city, the solar irradiation has a different effect on each building, which also depends on the hour of the day. In addition, this effect is highly non-linear, and therefore, the global horizontal irradiance is not a good predictor of the heat demand by itself.

Heat storage tanks and heat pumps can be combined to maintain high levels of comfort in a more cost-efficient way, and to avoid the over-sizing of the equipment. A daily analysis of the heat demand can help to set an upper bound to the size of the heat tank, while with the hourly analysis we can set an upper bound of the power of the heating equipment by observing the day and the hour of maximal heat demand respectively. The level of comfort, and the cost of each unit of comfort can be expressed as a function of the heat storage tank volume and the heat pump power in order to minimize the annualized cost of the selected level of comfort.

The control system of the temperature of the heat storage tanks plays a crucial role in maximizing the effectiveness of the heating system. Without an effective control algorithm it is not possible to size the heating equipment adequately. An improved control system based on a PID control was proposed as an easy to implement solution to control the temperature of the heat tanks. However, as many academic studies show, a MPC control can offer an even better control of the heat tank temperature and is also a good option to be implemented in CitySim.



## 5 References

- [ROBI11] Robinson, D., “Computer modelling for sustainable urban design”, Earthscan: London, pp. 121, 2011.
- [IEA12] Energy Policies of IEA Countries, Switzerland; International Energy Agency, 2012 Online. [www.iea.org/publications/freepublications/publication](http://www.iea.org/publications/freepublications/publication), (last time accessed: 10/07/2016)
- [PRIV10] Prívar, S., Široký, J., Ferkl, L. & Cigler, J. “Model predictive control of a building heating system: The first experience”, Elsevier: Prague and Pilsen, pp. 564, 2010.
- [KARN09] Karnouskos, S. & Nass de Holanda, T. “Simulation of a Smart Grid City with Software Agents”, Third UK Sim European Symposium on Computer Modeling and Simulation, pp. 1, 2009.
- [KEIR12] Keirstead, J., Jennings, M. & Sivakumar, A. “A review of urban energy system models: Approaches, challenges and opportunities”, Elsevier: London, pp. 3851-3852, 2012.
- [PERE11] Perez, D., Kämpf, J.H., Wilke, U. & Robinson, D. “CitySim simulation: the case study of Alt-Wiedikon, a neighborhood of Zürich city”, CISBAT: Lausanne, pp. 3, 2011.
- [PERE13] Perez, D., Kämpf, J.H. & Scartezzini, J. “Urban Area Energy Flow Microsimulation for Planning Support: a Calibration and Verification Study”, International Journal on Advances in Systems and Measurements: Lausanne, pp. 264, 2013.
- [SITG16] SITG, Online: <http://ge.ch/sitg/>, (last time accessed: 10/07/2016)
- [REGB16] Federal Register of Buildings and Dwellings, Online: <http://www.bfs.admin.ch/bfs/portal/en/index/themen/00/05/blank/01.html>, (last time accessed: 10/07/2016)
- [PERE14] Perez, D. “A framework to model and simulate the disaggregated energy flows supplying buildings in urban areas”, École Polytechnique Fédérale de Lausanne, 2014.
- [ABIY15] Abiyeb R. H. & Tunay, M. “Optimization of High-Dimensional Functions through Hypercube Evaluation”, Computational Intelligence and Neuroscience: Lefkosa, 2015.
- [CATA12] Catalina, T., Iordache, V., Caracaleanu, B. “Multiple regression model for fast prediction of the heating demand”, Elsevier, Romania, 2012.
- [NUMB16] Numbeo, Online: [http://www.numbeo.com/property-investment/city\\_result.jsp?country=Switzerland&city=Geneva](http://www.numbeo.com/property-investment/city_result.jsp?country=Switzerland&city=Geneva), (last time accessed: 10/07/2016)
- [HPPG16] Heat pump price guides, Online: <http://www.heatpumppriceguides.com/>, (last time accessed: 10/07/2016)
- [SRWT16] Stainless rain water tanks, Online: <http://www.stainlessrainwatertanks.com/tank-prices>, (last time accessed: 10/07/2016)
- [ET16] Engineering toolbox, Online: [http://www.engineeringtoolbox.com/heat-loss-open-water-tanks-d\\_286.html](http://www.engineeringtoolbox.com/heat-loss-open-water-tanks-d_286.html), (last time accessed: 10/07/2016)
- [ALPI16] Alpiq, Online: <http://www.alpiq.ch/en/media-target-group/dossiers/electricity-price.jsp>, (last time accessed: 10/07/2016)
- [BEND15] Bendig, M., „Integration of Organic Rankine Cycles for Waste Heat Recovery in Industrial Processes“, École Polytechnique Fédérale de Lausanne, pp. 124, 2015.
- [JIN14] Jin, X., Maguire, J., & Christensen, D., „Model Predictive Control of Heat Pump Water Heaters for Energy Efficiency“, 2014 ACEEE Summer Study on Energy Efficiency in Buildings, Pacific Grove, 2014.



- [ISLA13] Islam, M. & Salam, A., “Modelling and Control System design to control Water temperature in Heat Pump“, Karlstads University, pp. 25, 2013.
- [WASE13] Waseem, M., “Advanced control strategies for optimal operation of a combined solar and heat pump system“, Loughborough University’s Institutional Repository, 2013.





## 6 Acknowledgements

Thanks to Dr Jérôme Kämpf for his invaluable advice and supervision of the thesis and the conference paper.

I would like to thank to Prof. François Maréchal for his advice and his interesting suggestions regarding the path to follow.

Thanks to Prof. Ignacio Linares Hurtado for his final evaluation and feed-back.

Particular thanks to Nahid Mohajeri for providing the real data of the heat demands in the Junction district that were used for the calibration of the model. The source of these data is SITG and UniGE.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL

---



## Annex I: Conference paper

### Massive 3D models and physical data for building simulation at the urban scale: a focus on Geneva and climate change scenarios

J.R. Vázquez-Canteli<sup>1</sup>, J.H. Kämpf<sup>1</sup>

<sup>1</sup>*Solar Energy and Building Physics Laboratory (LESO-PB),  
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland*

#### Abstract

In a time when governments are pursuing increasingly ambitious energy efficiency goals, building energy simulation at the urban scale, combined with the high availability of physical data, can help in performing this task. The purpose of this study is to describe the methodology used to create a geometrical 3D and physical model for the city of Geneva, and to analyze the energy-saving potential of several refurbishment measures, including the installation of more solar PV panels, for different climatic scenarios. Different sources of data are used to build a PostgreSQL database with the physical and geometrical characteristics of the buildings. The virtual models are simulated with CitySim, a program that estimates the energy demand for heating, cooling, and lighting of every building on an hourly basis. The models include, but are not limited to, the shadowing effect among the buildings, the passive solar gains through glazing, the internal gains due to occupants' activities, the solar reflectance and infrared emissivity of the surfaces, and the materials that provide the buildings with insulation from the outdoors. This research will also provide some results that can be used by the municipal authorities of Geneva as a reference to quantify the decrease of the energy demand that they can expect to achieve in the next years.

*Keywords: Residential energy use, sustainable development, CO<sub>2</sub> reduction strategies, renewable energy technologies, building physics, cities, solar PV.*

#### 1 Introduction

Switzerland is taking bold decisions to gradually phase out nuclear power and to reduce their greenhouse gas emissions by 20% by 2020. Since moving towards an energy mix in which the baseload generation will be satisfied by gas-fired power plants seems to be a good alternative [1], further efforts must be made in order to reduce CO<sub>2</sub> emissions. Most of Switzerland's energy-related CO<sub>2</sub> emissions come from transportation and space heating. Therefore, new measures towards a more efficient use of energy should be mainly adopted in these two areas.

National energy standards (such as SIA 380/1:2009 inspired by a MINERGIE label) aim to decrease the heat demand in buildings by providing strict limits in the energy consumption for new and retrofitted buildings. In addition, many boilers are being replaced by heat pumps or by more efficient energy conversion units. Also, the installation of solar PV-T systems will help to supply DHW and to reduce the dependency of many buildings on the grid. More effective HVAC controllers are also a promising approach to reduce the energy needs in the buildings [2], and electric vehicles will possibly be used as energy storage systems connected to the grid in the near future. For all of these reasons, the cities' energy systems will increase in complexity, whereas the dependency of the heating and cooling demands on the ambient temperature will weaken. Instead,



the energy demand will depend more on stochastic behaviors such as solar gains and occupancy profiles. Building simulation at the urban scale is a powerful tool to predict the behavior of the cities of tomorrow, allowing regulators to develop more effective policies to manage the energy supply and demand of the buildings on an urban scale [3] [4].

Urban simulation can help Geneva municipal authorities to quantify the decrease in energy demand that they can expect to achieve in their city thanks to different refurbishment scenarios. Climate change estimations provided by the Meteororm software are helpful to evaluate the expected effect of global warming on the heating and cooling needs of the city. Some models based on PostgreSQL databases and GIS have been used before to model other cities in Switzerland [5] [6].

This paper describes the methodology to realize the geometrical 3D and physical model for the city of Geneva, analyzes the energy-saving potential of different refurbishment measures, and provides some useful information about the total energy generation that could be achieved thanks to the installation of more PV panels. The results will be based on a particular district of Geneva.

## 2 Methodology

In order to create a geometrical 3D and physical model that we can simulate for solving the energy flows, it is necessary to store that information in a database with accurate data of the buildings.

### 2.1 Creation of the database

PostgreSQL is a complete DBMS which allows a wide range of SQL functionalities. The PostGIS module provides additional geometry types and functions to handle them. The database was completed with data from the information system of the territory of Geneva (SITG) [7] and from the Federal Register of Buildings and Dwellings (RegBL) [8]. Figure 1 depicts graphically the structure of the database with a unique identification number or EGID assigned to each building, all pointing to a given district, scenario to be simulated, and occupancy type (i.e. residential, commerce). The grounds of the city and the walls of the buildings were defined as composites, where each composite is formed by several layers, and each layer is made of a specific material.

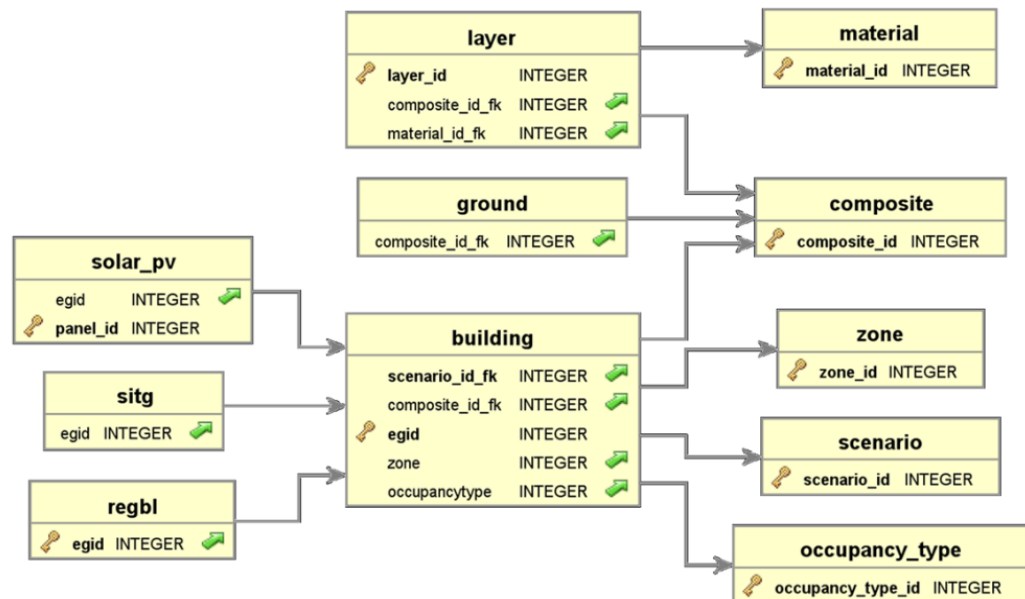


Figure 1: EV diagram of the database.



The year of construction of each building is useful to estimate the composition of the walls and the roofs, the natural ventilation rates, the characteristics of the windows, and the glazing ratios based on an existing study [9], as table 1 shows.

Data of the heights, number of floors, number of occupants, type of building, and year of construction and refurbishment were obtained from SITG and RegBL, while the information about the footprint geometries was only available in SITG. For some buildings, the height and the number of floors were missing. In these cases (0.54% of the buildings) these parameters were estimated using the average values of the buildings in the same district. When the period of construction was not available (1% of the buildings), it was estimated using a Monte Carlo approximation based on the age of the buildings located in the same district.

Table 1: Composition of the walls for buildings built between 1900 and 1945 [9].

$U_{wall}$ ( $W/m^2K$ )	Layer material	Thick. (m)	$\lambda$ ( $W/m \cdot K$ )	$C_p$ ( $J/kg \cdot K$ )	Density ( $kg/m^3$ )
0.94	Insulating rendering	0.02	0.08	1000	300
	Rubble masonry	0.40	0.81	1045	1600
	Insulating plaster	0.02	0.21	800	900

The windows were defined in the model using the U-value of the glass, eqn. (1), the g-value, and the glazing ratio with respect to the wall, as shown in table 2.

Table 2: Parameters used to model the windows.

Period	U-value ( $W/m^2 \cdot K$ )	g-value	Glazing ratio	Window to wall ratio
Before 2000	2.3	0.47	0.175	0.25
After 2000	1.7	0.49	0.245	0.35

$$U_{glass} = \frac{fenestration\ ratio}{glazing\ ratio} \cdot U_{window} \quad (1)$$

The type of building is useful to estimate the occupancy and the electricity consumption profiles, the minimum ventilation rates required, and the number of occupants based on the values of the SIA2024 norm, as shown in table 3. The number of occupants of each building was calculated as the average between the value provided by SITG and RegBL, and the value estimated from the norm. If the ratio between these two values was greater than 50% of the value in the SIA2024, only the value from the norm was used.

Table 3: Values based on the SIA2024 for residential buildings and offices.

Type of building	Surface/ pers. ( $m^2$ )	Electric pow. ( $W/m^2$ )	Ventilation day ( $m^3/m^2 \cdot h$ )	Ventilation night ( $m^3/m^2 \cdot h$ )
Residential	50	2	0.6	0.3
Office	14	7	2.6	0

The shortwave reflectance of all the walls was set to 25%, and the indoor minimum and maximum comfort temperatures were set to 21.0°C and 26.0°C.

## 2.2 Creation of the geometrical 3D and physical model

After creating a database, a java program was used to write an XML file that provides the simulation software CitySim with all the information of the physical and geometrical characteristics of the buildings. CitySim was developed at EPFL [10], and computes an estimation of the energy demand for heating, cooling, and lighting of every building. After the simulation, CitySim returns a text file with all the results for every floor and building on an hourly basis. These results are put into the database and linked to each building by the EGID number. This allows analysis of the results in combination with the physical characteristics of the buildings. A geometrical 3D and physical model of one district of Geneva was created (fig. 2) in order to perform a calibration with real data of the heat demands, and to analyze the behavior of the buildings in different scenarios.

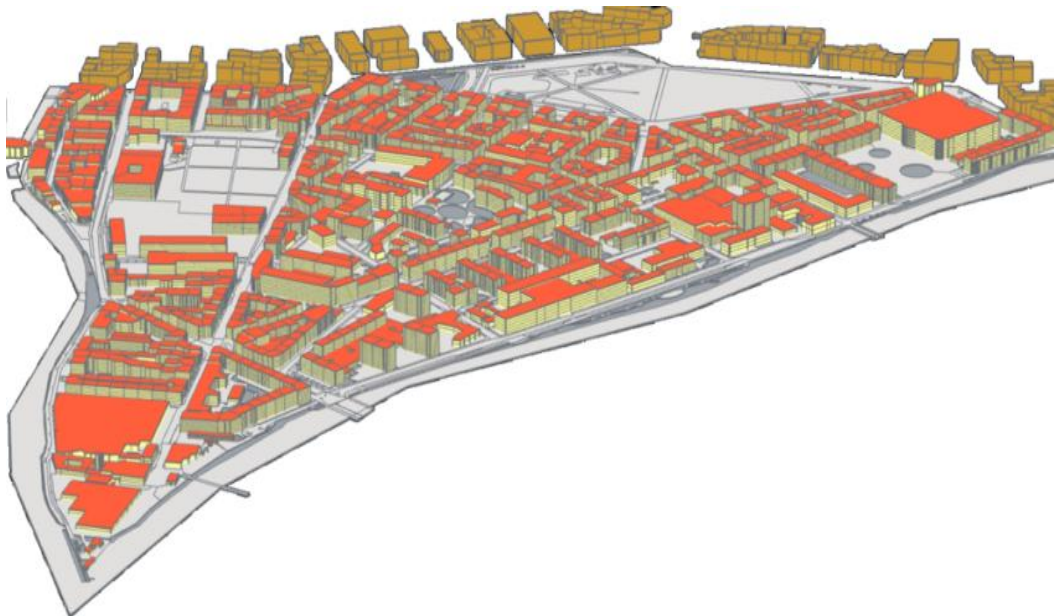


Figure 2: Geometrical 3D and physical model of the Junction district in Geneva.

## 2.3 Calibration of the model

In order to calibrate the model, the buildings were classified based on their period of construction and their type of use. These two features have a great impact on the total ventilation rate of the buildings, which is difficult to estimate and can change the heating and cooling demands dramatically. The values of the minimum ventilation requirements were available in the SIA2024 norm, while the initial natural ventilation rates were obtained from a previous study [9]. Table 4, and table 5 show the median value of the  $\log_2$  of the quotient between the simulated heat demand ( $Q_{sim}$ ) and the actual value ( $Q_{real}$ ), which is used to measure the error. The number of buildings is expressed between brackets.

Table 4: Results from the calibration for different periods of construction.

Period	Before calibration		After calibration	
	$\eta_{nat} (h^{-1})$	$\log_2(Q_{sim}/Q_{real})$	$\eta_{nat} (h^{-1})$	$\log_2(Q_{sim}/Q_{real})$
<1915 (192)	0.700	-0.410	1.078	-0.034
1915-1945 (136)	0.700	-0.526	1.179	-0.051
1946-1960 (79)	0.600	-0.743	1.224	-0.007
1961-1970 (79)	0.550	-0.692	1.300	0.130
1971-1980 (29)	0.500	-0.696	1.191	0.074
1981-1990 (42)	0.400	-1.080	1.093	-0.040
1991-2000 (34)	0.350	-0.961	0.920	-0.010
>2000 (24)	0.300	-0.434	0.599	-0.005

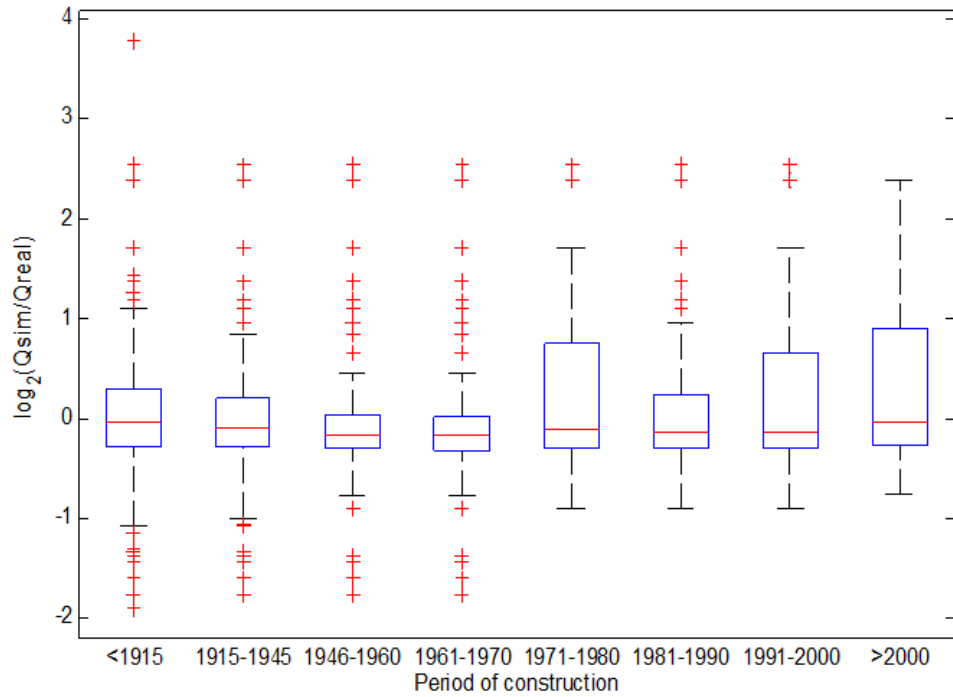


Figure 3: Error after the calibration for different periods of construction.

The total ventilation rate was calculated using eqn. (3), where  $\gamma_{\eta_{nat}}$  and  $\gamma_{vent}$  are two multiplier coefficients adjusted during the calibration process (table 5). Before the calibration, these multipliers had a value of one.

$$Ventilation = \max\{ \gamma_{\eta_{nat}} \cdot \eta_{nat} , \gamma_{vent} \cdot Ventilation_{SIA2024} \} \quad (3)$$

Table 5: Results from the calibration for different types of buildings.

Type of building	Before calibration	After calibration		
	$\log_2(Q_{sim}/Q_{real})$	$\gamma_{nat}$	$\gamma_{vent}$	$\log_2(Q_{sim}/Q_{real})$
Commercial (42)	0.349	0.525	0.600	-0.024
Education (30)	0.002	0.542	0.628	-0.025
Residential (442)	-0.733	1.067	1.737	-0.020
Office (45)	-0.074	0.605	0.669	0.112
Hotel (3)	-0.466	0.970	1.591	0.011
Restaurant (3)	0.273	1.057	0.815	-0.089
Other (50)	0.134	0.440	0.360	-0.034

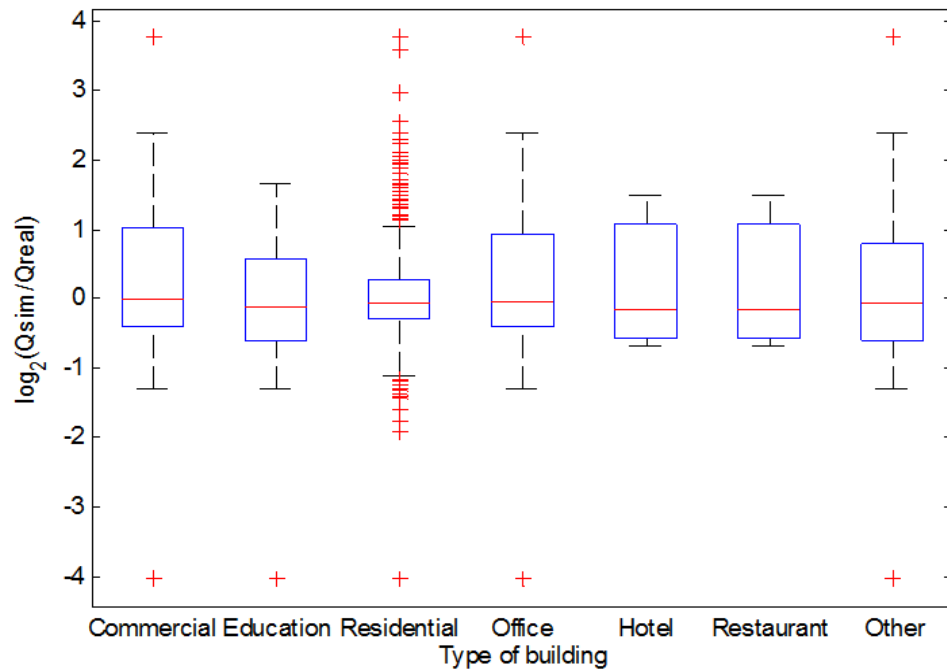


Figure 4: Error after the calibration for different types of buildings.

### 3 Results and discussion

From the calibration, we can observe that the heat demand of 50% of the residential buildings has a margin of error of about  $\pm 15\%$ , as shown in fig. 4. Once the model was calibrated, several refurbishment and climate change scenarios were analyzed. These analyses can be used by the municipal authorities of Geneva as a reference to quantify the decrease of the energy demand that they can expect to achieve in the next years if they adopt similar measures.



In the low refurbishment scenario, all the windows are triple glazed ( $U_{\text{window}} = 0.87 \text{ W/m}^2\cdot\text{K}$  and  $g_{\text{window}} = 0.6$ ), while the high refurbishment includes triple glazed windows, increased wall insulations ( $U_{\text{wall}} = 0.2 \text{ W/m}^2\cdot\text{K}$ ) and lower natural ventilation rates ( $\eta_{\text{nat}} = 0.3 \text{ h}^{-1}$ ) in all the buildings. Furthermore, PV panels were installed on all the available surfaces of the roofs with a gross annual irradiation greater than  $1000 \text{ kWh/m}^2$ . The panels considered were SunPower X21-345, which have an efficiency of 21.5%. The heat pumps used in this model have a technical efficiency of 30%, a ground source with vertical pipes from 5m to 10m deep in a soil with a thermal diffusivity of  $0.07 \text{ m}^2/\text{day}$ , and a target temperature of  $55^\circ\text{C}$ .

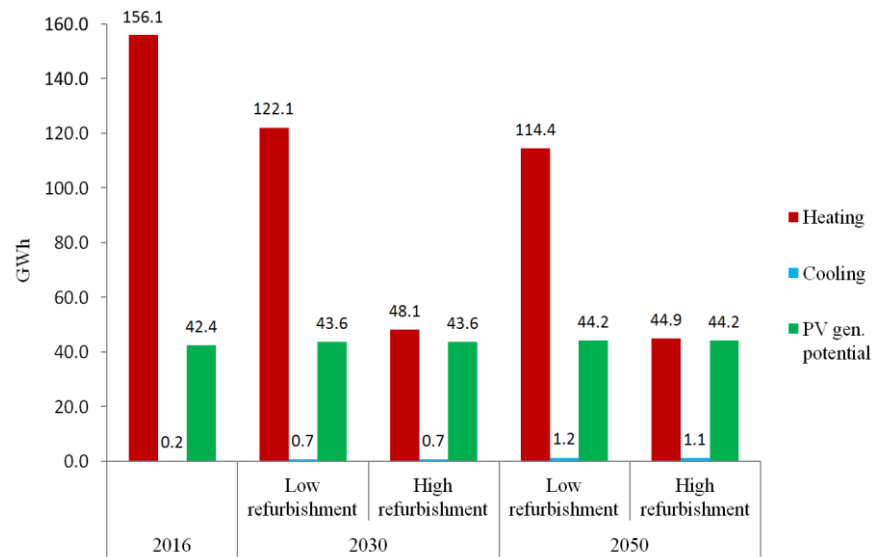


Figure 5: Energy demand and solar PV potential for different scenarios.

The heat demand of this district would decrease due, in part, to the effects of climate change, while the cooling demand will increase, as shown in fig. 5. However, in all the scenarios considered, the predicted cooling demand would still be too low to justify the wide implementation of cooling systems in this district. In 2050, increasing the insulation of the walls and reducing the infiltration rates would result in lower cooling demands of the buildings. Therefore, the better insulation of these buildings would reduce the effect of the outdoor temperature on the indoor temperature, more than it would increase the heating due to the internal gains.

It is remarkable how it would be possible to achieve similar values of heat demand and energy generation from solar PV if the buildings were deeply refurbished. In 2050, considering a high refurbishment scenario and the heat demand to be supplied only by heat pumps, the annual electricity generation of the buildings would exceed their electricity demand. Therefore, with these assumptions, it would be possible to achieve an annual surplus of electricity generation of 4.9 GWh if all the solar PV potential of the roofs were used. However, one of the limitations of this model is that, due to the lack of data about the geometries of the roofs, all the PV panels were oriented horizontally.

Solar PV panels will not always generate enough electricity to supply the whole demand of a building, whereas in other times the generation will surpass the demand. Therefore, it is interesting to analyze the average hourly surplus and shortage of electricity in the whole district (fig. 6).

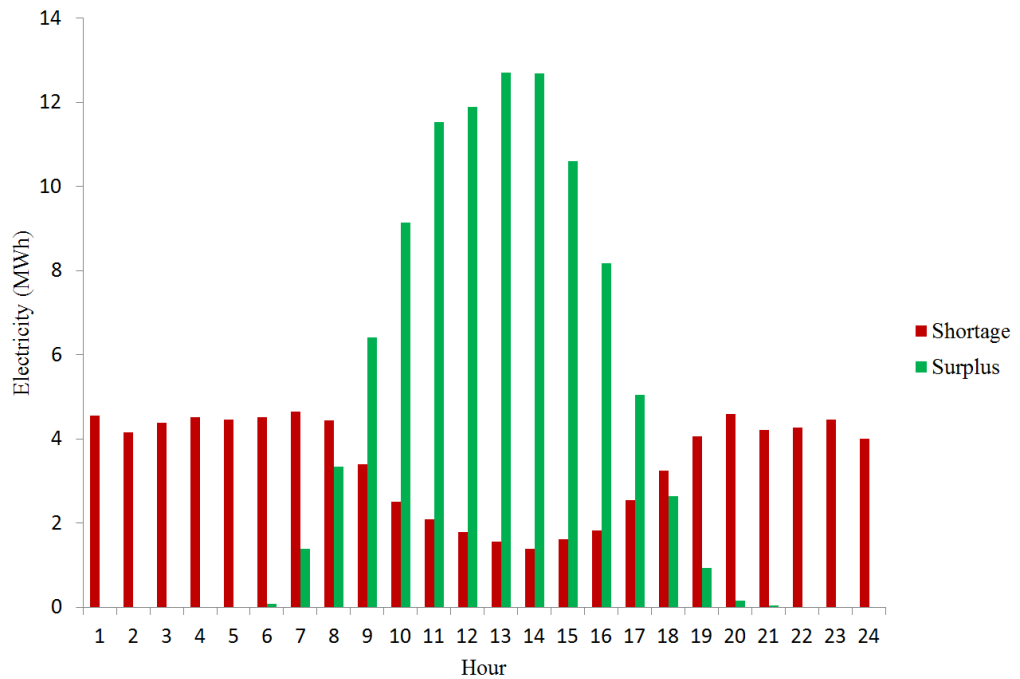


Figure 6: Estimation of the average hourly electricity shortage and surplus for the Junction district in 2050 and for a high refurbishment scenario.

During the daytime, from 9h to 17h, there is more electricity being generated than being consumed. However, there are still many buildings that, due to their low PV potential or high energy demand, are not independent from the grid during this period. A well-coordinated electrical distribution network could help in reducing the number of buildings whose electrical demand is not satisfied by solar PV panels. A maximum of 27.2 MWh of this electricity could be redistributed every day, on average, among the buildings of this same district that would need it during the daytime. The remaining 69.4 MWh of surplus energy could be stored in the buildings or in centralized systems, with the objective of using it during the periods of shortage. Heating up water to store it in tanks could be an option, while it is possible that part of this electricity could be used in the future to charge the batteries of electric vehicles.

High buildings with a small roof surface will probably need more energy than they will be able to generate, while buildings with a higher surface-height ratio will generate more than they will need (fig. 7).



Figure 7: Map of the estimated annual net electricity consumption in the Junction district for a high refurbishment scenario in 2050.

In this case, the net electricity consumption is not uniformly distributed. Some buildings, located on the west part of the district, have the potential to generate much more energy that they would need. Municipal authorities can take advantage of this potential, and adopt policies that promote the installation of centralized energy storage systems near these buildings. This measure, combined with the existing policies of building refurbishment and the installation of more PV panels on all the roofs, can drastically reduce the dependency of this district on non-renewable energy technologies.

## 4 Conclusion

Building simulation at the urban scale is a powerful tool to simulate the energy needs and generation potential of cities. PostgreSQL databases can store all the geometries of the buildings and their physical characteristics. Once the geometrical 3D and physical model is realized and calibrated, it can be simulated to obtain valuable information about the energy needs or generation of the buildings in different pre-defined scenarios. These data can be stored in the database together with the physical characteristics of the buildings for further analysis. For these models to be useful, the availability of accurate information regarding the physical characteristics of the buildings and their geometries is crucial. The results of the calibration show how it was possible to model the residential buildings accurately with the information that was available. However, it is more difficult to achieve this accuracy for other types of buildings, especially for those in which the stochastic behavior of the occupants is more unpredictable and has a greater impact on the heat demand.



The energy demand and generation of a district in Geneva were studied for different refurbishment scenarios, taking into account the expected effects of climate change in 2030 and 2050. These scenarios included an ideal case with the implementation of PV panels on every roof, and the use of heat pumps in every building. These measures were proven to drastically reduce the energy demands of the buildings, while increasing their generation of electricity above the overall annual energy consumption of the district. However, this potential energy surplus is not uniformly distributed. Some buildings have a much greater potential for installing PV panels than others, while certain buildings with smaller rooftop areas and higher heights would not be able to generate more electricity than they would consume. Municipal authorities of Geneva must take into account these asymmetries to adopt policies that promote an efficient distribution of PV panels, energy storage systems, and adequate interconnections in the energy distribution network.

Taking maximum advantage of the buildings with the highest solar potential, and installing centralized energy storage systems, such as water heat tanks, near them would be a possible good solution. In this case, a combination of PV panels and solar heaters would be a good approach. The stored heat can be redistributed through a district heating network or converted into electricity and distributed through the electrical power grid when the energy demand is higher. Further analysis can be developed with thermo-economic optimization techniques to find an optimal solution.

## 5 References

- [1] Energy Policies of IEA Countries, Switzerland; International Energy Agency, 2012 Online. [www.iea.org/publications/freepublications/publication](http://www.iea.org/publications/freepublications/publication)
- [2] Prívará, S., Šíroký, J., Ferkl, L. & Cigler, J. *Model predictive control of a building heating system: The first experience*, Elsevier: Prague and Pilsen, pp. 564, 2010.
- [3] Karnouskos, S. & Nass de Holanda, T. *Simulation of a Smart Grid City with Software Agents*, Third UK Sim European Symposium on Computer Modeling and Simulation, pp. 1, 2009.
- [4] Keirstead, J., Jennings, M. & Sivakumar, A. *A review of urban energy system models: Approaches, challenges and opportunities*, Elsevier: London, pp. 3851-3852, 2012.
- [5] Perez, D., Kämpf, J.H., Wilke, U. & Robinson, D. *CitySim simulation: the case study of Alt-Wiedikon, a neighborhood of Zürich city*, CISBAT: Lausanne, pp. 3, 2011.
- [6] Perez, D., Kämpf, J.H. & Scartezini, J. *Urban Area Energy Flow Microsimulation for Planning Support: a Calibration and Verification Study*, International Journal on Advances in Systems and Measurements: Lausanne, pp. 264, 2013.
- [7] SITG, <http://ge.ch/sitg/>
- [8] Federal Register of Buildings and Dwellings, <http://www.bfs.admin.ch/bfs/portal/en/index/themen/00/05/blank/01.html>
- [9] Perez, D. *A framework to model and simulate the disaggregated energy flows supplying buildings in urban areas*, École Polytechnique Fédérale de Lausanne, 2014.
- [10] Robinson, D., *Computer modelling for sustainable urban design*, Earthscan: London, pp. 121, 2011.

## 6 Acknowledgments

Particular thanks to Nahid Mohajeri for providing the real data of the heat demands in the Junction district that were used for the calibration of the model. The source of these data is SITG and UniGE.

The authors would like to acknowledge the advices of Prof. François Maréchal on the presented work.



## Annex II: MATLAB code to read and display the heat demand behavior as a function of the temperature and the solar irradiance

```
clear java
clear import
javaclasspath('postgresql-9.4.1207.jar','postgis-jdbc-
2.2.0dev.jar');

props=java.util.Properties;
props.setProperty('user', 'USER');
props.setProperty('password', 'PASSWORD');

driver=org.postgresql.Driver;
url = 'jdbc:postgresql://128.178.6.45:5432/geneva';
conn=driver.connect(url, props);

query='select egid,q_present,q_low,q_high from
thermal_behaviour_hourly_nopeople order by egid desc'; % Gets all
records
st=conn.createStatement();
rs=st.executeQuery(query);

i = 1;
while (rs.next())
    results1 =
    strsplit(char(rs.getString('q_present')),{'{','}',' ',' '});
    q_present(i,:) =
    100*str2double(results1(2:8761))/sum(str2double(results1(2:8761)))
    ;
    q_present_kwh(i,:) = str2double(results1(2:8761))/1000;

    results2 =
    strsplit(char(rs.getString('q_low')),{'{','}',' ',' '});
    q_low(i,:) =
    100*str2double(results2(2:8761))/sum(str2double(results1(2:8761)))
    ;
    q_low_kwh(i,:) = str2double(results2(2:8761))/1000;

    results3 =
    strsplit(char(rs.getString('q_high')),{'{','}',' ',' '});
    q_high(i,:) =
    100*str2double(results3(2:8761))/sum(str2double(results1(2:8761)))
    ;
    q_high_kwh(i,:) = str2double(results3(2:8761))/1000;
    i = i + 1
end

query='select egid,temperature_present,
temperature_low,temperature_high from thermal_behaviour_hourly
order by egid desc'; % Gets all records
```

```

st=conn.createStatement();
rs=st.executeQuery(query);

i = 1;
while (rs.next())
    results4 =
strsplit(char(rs.getString('temperature_present')),{' ',' ',' ',' '});
    Ta_present(i,:) = str2double(results4(2:8761));

    results5 =
strsplit(char(rs.getString('temperature_low')),{' ',' ',' ',' '});
    Ta_low(i,:) = str2double(results5(2:8761));

    results6 =
strsplit(char(rs.getString('temperature_high')),{' ',' ',' ',' '});
    Ta_high(i,:) = str2double(results6(2:8761));

    results0 = char(rs.getString('egid'));
    egid(i,1) = str2double(results0);

    i = i + 1;
end

query2='select
temperature_present,temperature_2030,irradiance_present,irradiance
_2030,hour from temperature_irradiance order by hour asc';
st2=conn.createStatement();
rs2=st2.executeQuery(query2);

i = 1;
while (rs2.next())
    results5 = char(rs2.getString('temperature_present'));
    temperture_irradiance(i,1) = str2double(results5);

    results6 = char(rs2.getString('temperature_2030'));
    temperture_irradiance(i,2) = str2double(results6);

    results7 = char(rs2.getString('irradiance_present'));
    temperture_irradiance(i,3) = str2double(results7);

    results8 = char(rs2.getString('irradiance_2030'));
    temperture_irradiance(i,4) = str2double(results8);

    results9 = char(rs2.getString('hour'));
    temperture_irradiance(i,5) = str2double(results9);

    i = i + 1;
end

for i=1:1:length(Ta_present(:,1))
    DT_present(i,:) = Ta_present(i,:)-temperture_irradiance(:,1)';
    DT_low(i,:) = Ta_low(i,:)-temperture_irradiance(:,2)';

```

```

    DT_high(i,:) = Ta_high(i,:)-temperture_irradiance(:,2)';
end

for i=1:1:365
    DT_present_day(:,i) = sum(DT_present(:,(i-1)*24+1:i*24)')'/24;
    DT_low_day(:,i) = sum(DT_low(:,(i-1)*24+1:i*24)')'/24;
    DT_high_day(:,i) = sum(DT_high(:,(i-1)*24+1:i*24)')'/24;
    q_present_day(:,i) = sum(q_present(:,(i-1)*24+1:i*24)')';
    q_low_day(:,i) = sum(q_low(:,(i-1)*24+1:i*24)')';
    q_high_day(:,i) = sum(q_high(:,(i-1)*24+1:i*24)')';
    q_present_day_kwh(:,i) = sum(q_present_kwh(:,(i-1)*24+1:i*24)')';
    q_low_day_kwh(:,i) = sum(q_low_kwh(:,(i-1)*24+1:i*24)')';
    q_high_day_kwh(:,i) = sum(q_high_kwh(:,(i-1)*24+1:i*24)')';
    daily_irr(i,1) = sum(temperture_irradiance((i-1)*24+1:i*24,3))/24;
    daily_irr(i,2) = sum(temperture_irradiance((i-1)*24+1:i*24,4))/24;
end

for i=1:1:823
    V(i,1) = 3600*max(q_present_day_kwh(i,:))/(1000*4.18*(35-21));
    V(i,2) = 3600*max(q_low_day_kwh(i,:))/(1000*4.18*(35-21));
    V(i,3) = 3600*max(q_high_day_kwh(i,:))/(1000*4.18*(35-21));
end

for i=1:1:length(DT_present(:,1))
    mdl1 = fitlm(DT_present_day(i,:)','q_present_day(i,:));
    R2(i,1) = mdl1.Rsquared.Adjusted;
    mdl2 = fitlm(DT_low_day(i,:)','q_low_day(i,:));
    R2(i,2) = mdl2.Rsquared.Adjusted;
    mdl3 = fitlm(DT_high_day(i,:)','q_high_day(i,:));
    R2(i,3) = mdl3.Rsquared.Adjusted;
end

binranges = [min(min(V)):(max(max(V))-min(min(V)))/100:max(max(V))];
color = {'b','r','g'};%[0 1 1;1 0 1;1 1 0]
%ax=subplot(2,1,1)
for i=1:1:3
    [bincounts,xout] = hist(V(:,i),binranges)
    B=bar(xout,100*bincounts/823,color{i})
    ch = get(B,'child');
    set(ch,'facea',1-(i-1)*0.25)
    set(ch,'EdgeColor',color{i})
    h(i,:) = B;
    hold on
    [AX,H1,H2] =
plotyy(xout,zeros(1,length(xout)),xout,cumsum(xout.*bincounts),'plot')
    set(H2,'Color',color{i});
    h(i+3,:) = H2;
    hold on

```

```

        set (AX(1), 'YLim', [0 16])
        set (AX(1), 'YTick', [0:2:16])
        set (AX(2), 'YLim', [0 80000])
        set (AX(2), 'YTick', [0:10000:80000])
        set (AX(1), 'XLim', [0-(max(max(V))-min(min(V)))/200
max(max(V))] )
        set (AX(1), 'XTick', [0:50:max(max(V))] )
        set (AX(2), 'XLim', [0-(max(max(V))-min(min(V)))/200
max(max(V))] )
        set (AX(2), 'XTick', [0:50:max(max(V))] )
end
hold off
xlabel('Heat tank volume [m^3]')
ylabel(AX(1), '% of buildings')
ylabel(AX(2), 'Total cumulative volume [m^3]')
legend(h, 'No refurbishment 2016', 'Low refurbishment 2030', 'High
resurbishment 2030')
title('Heat tanks size distribution and cumulative storage volume
(823 buildings)')

ax=subplot(2,1,1)
binranges = [0.8:0.002:1.0];
color = {'b', 'r', 'g'};%[0 1 1;1 0 1;1 1 0]
for i=1:1:3
    [bincounts,xout] = hist(R1(:,i),binranges)
    B=bar(ax,xout,100*bincounts/823,color{i})
    ch = get(B, 'child');
    set(ch, 'facea', 1-(i-1)*0.25)
    set(ch, 'EdgeColor', color{i})
    hold on
    %set(get(gca, 'child'), 'FaceColor', color(i,:), 'EdgeColor', 'k');
end
hold off
xlim([0.8 1]);
ylim([0 20]);
xlabel('R^2_a_d_j_u_s_t_e_d')
ylabel('% of buildings')
set(ax, 'XTick', [0.75:0.02:1])
legend('No refurbishment 2016', 'Low refurbishment 2030', 'High
resurbishment 2030')
title('Distribution of R^2_a_d_j_u_s_t_e_d (model without
occupants)')

%title('Distribution of R^2_a_d_j_u_s_t_e_d (regressors: \DeltaT,
Global horizontal irradiance)')

ax=subplot(2,1,2)
binranges = [0.8:0.0025:1.0];
for i=1:1:3
    [bincounts,xout] = hist(R2(:,i),binranges)
    %ax=subplot(2,1,2)
    B=bar(ax,xout,100*bincounts/823,color{i})
    ch = get(B, 'child');
    set(ch, 'facea', 1-(i-1)*0.25)

```



```

        set(ch, 'EdgeColor', color{i})
        hold on
        %set(get(gca, 'child'), 'FaceColor', color(i,:), 'EdgeColor', 'k');
    end
    hold off
    xlim([0.8 1]);
    ylim([0 25]);
    xlabel('R^2_a_d_j_u_s_t_e_d')
    ylabel('% of buildings')
    set(ax, 'XTick', [0.75:0.02:1])
    legend('No refurbishment 2016', 'Low refurbishment 2030', 'High
resurbishment 2030')
    title('Distribution of R^2_a_d_j_u_s_t_e_d (model without
occupants)')

    maximum = rem(find(R2-max(max(R2))==0), 823);
    minimum = rem(find(R2-min(min(R2))==0), 823);
    j=1;
    commercial=[2037107 2038019 295094870]
    for i=609%i=minimum:maximum-minimum:maximum
        %Figure(i+1)
        ax=subplot(1,2,j)

        scatter3(ax,DT_present_day(i,:),daily_irr(:,1),q_present_day(i,:),
'b')
        hold on
        scatter3(ax,DT_low_day(i,:),daily_irr(:,2),q_low_day(i,:), 'r')
        hold on

        scatter3(ax,DT_high_day(i,:),daily_irr(:,2),q_high_day(i,:), 'g')
        xlabel('\DeltaT [°C]')
        zlabel('Qh [^o/oo]')
        ylabel('Global horizontal irradiance [W/m^2]')
        legend('No refurbishment 2016', 'Low refurbishment 2030', 'High
resurbishment 2030')
        title(sprintf('Heat demand behaviour for different
refurbishment scenarios in 2D\nExpressed as a ^o/oo of Qh_2_0_1_6
'))
        ax=subplot(1,2,j+1)
        scatter(ax,DT_present_day(i,:),q_present_day(i,:), 'b')
        hold on
        scatter(ax,DT_low_day(i,:),q_low_day(i,:), 'r')
        hold on
        scatter(ax,DT_high_day(i,:),q_high_day(i,:), 'g')
        xlabel('\DeltaT [°C]')
        ylabel('Qh [^o/oo]')
        legend('No refurbishment 2016', 'Low refurbishment 2030', 'High
resurbishment 2030')
        title(sprintf('Heat demand behaviour for different
refurbishment scenarios in 2D\nExpressed as a ^o/oo of Qh_2_0_1_6
'))
        j=j+2;
    end
    hold off

```





## Annex III: MATLAB code to optimize the calibration using the Hypercube optimization method.

```
clear java
clear import
javaclasspath('postgresql-9.4.1207.jar','postgis-jdbc-2.2.0dev.jar');
%HYPERCUBE OPTIMIZATION ALGORITHM
%Definition of the parameters of the algorithm
N = 75;
M = 28;
alpha = 0.5;
q1 = 0.60;
q2 = 0.40;
movCount = 0;
margin = 0.5;
infiltrationRate0 = 1.3*[0.7 0.7 0.6 0.55 0.5 0.4 0.35 0.3];
natVentCoeff0 = zeros(1,10) + 1;
ventCoeff0 = zeros(1,10) + 1;

infiltrationRateRef = 1.3*[0.7 0.7 0.6 0.55 0.5 0.4 0.35 0.3];

natVentCoeffUB = natVentCoeff0*(1 + margin);
ventCoeffUB = ventCoeff0*(1 + margin);
infiltrationRateUB = infiltrationRate0*(1 + margin);
natVentCoeffLB = natVentCoeff0*(1 - margin);
ventCoeffLB = ventCoeff0*(1 - margin);
infiltrationRateLB = infiltrationRate0*(1 - margin);

%-----*-----INITIALIZATION-----
UB(1,:) = [natVentCoeffUB ventCoeffUB infiltrationRateUB];
LB(1,:) = [natVentCoeffLB ventCoeffLB infiltrationRateLB];
Xc(1,:) = (UB(1,:) + LB(1,))/2;
Ro(1,:) = (UB(1,:) - LB(1,))/2;
S = zeros(M,length(UB(1,)))+1;
Fbest(1) = 9999999;
%Tolerances
tolF = 0.005;
toldnn = 0.01;
countD = 0;
countF = 0;
countTol = 0;

natVentCoeffc = Xc(1,1:length(natVentCoeffUB));
ventCoeffc =
Xc(1,length(natVentCoeffUB)+1:(length(natVentCoeffUB)+length(ventC
oeffUB)));
infiltrationRatec =
Xc(1,length(natVentCoeffUB)+length(ventCoeffUB)+1:(length(infiltra
tionRateUB)+length(natVentCoeffUB)+length(ventCoeffUB)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

n = 1;
A =
textread('CitysimDatabaseLinkerXML.java','%s','delimiter','\n',
'bufsize',2147483647);

A{7} = sprintf('public class CitySimDatabaseLinkerXML%d',n);
A{150} = sprintf('double[] natVentilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};',natVentCoeffc(1),natVentCoeffc(2)
),natVentCoeffc(3),natVentCoeffc(4),natVentCoeffc(5),natVentCoeffc(
6),natVentCoeffc(7),natVentCoeffc(8),natVentCoeffc(9),natVentCoef
fc(10));
A{151} = sprintf('double[] ventilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};',ventCoeffc(1),ventCoeffc(2),vent
Coeffc(3),ventCoeffc(4),ventCoeffc(5),ventCoeffc(6),ventCoeffc(7),
ventCoeffc(8),ventCoeffc(9),ventCoeffc(10));
A{61} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.16 WHERE gbauj <
1915;");',infiltrationRatec(1));
A{63} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value =1.16 WHERE gbauj BETWEEN
1915 AND 1945;");',infiltrationRatec(2));
A{65} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj BETWEEN
1946 AND 1960;");',infiltrationRatec(3));
A{67} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj BETWEEN
1961 AND 1970;");',infiltrationRatec(4));
A{69} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.84 WHERE gbauj BETWEEN
1971 AND 1980;");',infiltrationRatec(5));
A{71} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.43 WHERE gbauj BETWEEN
1981 AND 1990;");',infiltrationRatec(6));
A{73} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.29 WHERE gbauj BETWEEN
1991 AND 2000;");',infiltrationRatec(7));
A{75} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.21 WHERE gbauj >
2000;");',infiltrationRatec(8));

name = sprintf('CitysimDatabaseLinkerXML%d.java',n);
[fileID, message] = fopen(name,'wt+');
for k = 1:1:length(A)
    fprintf(fileID,'%s\n',A{k});
end
fclose(fileID);

command = sprintf('mingw32-make -f MakefileXML.mingw32
MAINCLASS=CitySimDatabaseLinkerXML%d run',n);
[status,cmdout] = system(command);
command = 'CitySimPro essai.xml';
[status,cmdout] = system(command);
command = sprintf('mingw32-make -f MakefileReader.mingw32 run');
[status,cmdout] = system(command);

```

```

props=java.util.Properties;
props.setProperty('user', 'USER');
props.setProperty('password', 'PASSWORD');

% Create the database connection (port 5432 is the default
postgres chooses
% on installation)
driver=org.postgresql.Driver;
url = 'jdbc:postgresql://128.178.6.45:5432/geneva';
conn=driver.connect(url, props);

% A test query
query='select 0.5*sum(((heat_supply-
heat_supply_sim)/heat_supply)^2)/count(*) as sme from calibration
where scenario_id_fk = 0'; % Gets all records
st=conn.createStatement();
rs=st.executeQuery(query);
rs.next();
smelc = str2double(char(rs.getString('sme')));
sme2c = 0.5*(sum([1 1 1 1 1 1 1 1 1 1]-natVentCoeffc).^2) +
sum([1 1 1 1 1 1 1 1 1 1]-ventCoeffc).^2) +
1.5*sum(((infiltrationRateRef-
infiltrationRatec)/infiltrationRateRef).^2))/3.5;
Fc(1) = q1*smelc + q2*sme2c;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fbest(1) = 9999999999;
Xbest(1,:) = Xc(1,:);
dnn(1,:) = 1;
%-----UPDATING PARAMETERS-----
for j = 1:1:M %M iterations or displacements of the hypercube
%Initialize values
2
if j>1
Fbest(j) = Fbest(j-1);
Xbest(j,:) = Xbest(j-1,:);
end
parfor i = 1:1:N
for k = 1:1:length(UB(1,:))
X(i,k) = unifrnd(LB(j,k),UB(j,k),1,1); %Generating
random points uniformly distributed within the boudaries
end

natVentCoeff = X(i,1:length(natVentCoeffUB));
ventCoeff =
X(i,length(natVentCoeffUB)+1:(length(natVentCoeffUB)+length(ventCo
effUB)));
infiltrationRate =
X(i,length(natVentCoeffUB)+length(ventCoeffUB)+1:(length(infiltrat
ionRateUB)+length(natVentCoeffUB)+length(ventCoeffUB)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 2;

```

```

B =
textread('CitysimDatabaseLinkerXML.java', '%s', 'delimiter', '\n',
'bufsize', 2147483647);

B{7} = sprintf('public class
CitySimDatabaseLinkerXML%d', n);
B{150} = sprintf('double[] natVentilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};', natVentCoeff(1), natVentCoeff(2),
natVentCoeff(3), natVentCoeff(4), natVentCoeff(5), natVentCoeff(6), na
tVentCoeff(7), natVentCoeff(8), natVentCoeff(9), natVentCoeff(10));
B{151} = sprintf('double[] ventilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};', ventCoeff(1), ventCoeff(2), ventCo
eff(3), ventCoeff(4), ventCoeff(5), ventCoeff(6), ventCoeff(7), ventCoe
ff(8), ventCoeff(9), ventCoeff(10));
B{61} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 0.16 WHERE gbauj <
1915;");', infiltrationRate(1));
B{63} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj
BETWEEN 1915 AND 1945;");', infiltrationRate(2));
B{65} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj
BETWEEN 1946 AND 1960;");', infiltrationRate(3));
B{67} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj
BETWEEN 1961 AND 1970;");', infiltrationRate(4));
B{69} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 0.84 WHERE gbauj
BETWEEN 1971 AND 1980;");', infiltrationRate(5));
B{71} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 0.43 WHERE gbauj
BETWEEN 1981 AND 1990;");', infiltrationRate(6));
B{73} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 0.29 WHERE gbauj
BETWEEN 1991 AND 2000;");', infiltrationRate(7));
B{75} = sprintf('ps = c.prepareStatement("UPDATE building
SET infiltration_rate = %d, roof_u_value = 0.21 WHERE gbauj >
2000;");', infiltrationRate(8));

name = sprintf('CitysimDatabaseLinkerXML%d.java', n);
[fileID, message] = fopen(name, 'wt+');
for k = 1:length(B)
    fprintf(fileID, '%s\n', B{k});
end
fclose(fileID);

command = sprintf('mingw32-make -f MakefileXML.mingw32
MAINCLASS=CitySimDatabaseLinkerXML%d run', n);
[status, cmdout] = system(command);
command = 'CitySimPro essai.xml';
[status, cmdout] = system(command);
command = sprintf('mingw32-make -f MakefileReader.mingw32
run');
[status, cmdout] = system(command);

```

```

        % A test query
        query='select 0.5*sum(((heat_supply-
heat_supply_sim)/heat_supply)^2)/count(*) as sme from calibration
where scenario_id_fk = 0'; % Gets all records
        st=conn.createStatement();
        rs=st.executeQuery(query);
        rs.next();
        sme1(i,j) = str2double(char(rs.getString('sme')))
        sme2(i,j) = 0.5*(sum([1 1 1 1 1 1 1 1 1 1]-
natVentCoeff).^2) + sum([1 1 1 1 1 1 1 1 1 1]-ventCoeff).^2) +
1.5*sum(((infiltrationRateRef-
infiltrationRate)/infiltrationRateRef).^2))/3.5
        F(j,i) = q1*sme1(i,j) + q2*sme2(i,j)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %F(j,i) = sum(sme2.^2); %F for j=N+1 is Fmean
        if Fbest(j)>F(j,i)
            Fbest(j) = F(j,i)
            Xbest(j,:) = X(i,:);
            command = sprintf('copy
CitySimDatabaseLinkerXML%d.java Best_file /Y',n);
            [status,cmdout] = system(command);
        end
        i
    end
    Xmean(j,:) = ((1-alpha)*Xc(j)+alpha*Xbest(j,:));

    natVentCoeffmean = Xmean(j,1:length(natVentCoeffUB));
    ventCoeffmean =
Xmean(j,length(natVentCoeffUB)+1:(length(natVentCoeffUB)+length(ve
ntCoeffUB)));
    infiltrationRateMean =
Xmean(j,length(natVentCoeffUB)+length(ventCoeffUB)+1:(length(infil
trationRateUB)+length(natVentCoeffUB)+length(ventCoeffUB)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    n = 3;
    C =
textread('CitysimDatabaseLinkerXML.java','%s','delimiter','\n',
'bufsize',2147483647);

    C{7} = sprintf('public class CitySimDatabaseLinkerXML%d',n);
    C{150} = sprintf('double[] natVentilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};',natVentCoeffmean(1),natVentCoeff
mean(2),natVentCoeffmean(3),natVentCoeffmean(4),natVentCoeffmean(5
),natVentCoeffmean(6),natVentCoeffmean(7),natVentCoeffmean(8),natV
entCoeffmean(9),natVentCoeffmean(10));
    C{151} = sprintf('double[] ventilationCoeff =
{%d,%d,%d,%d,%d,%d,%d,%d,%d,%d};',ventCoeffmean(1),ventCoeffmean(2
),ventCoeffmean(3),ventCoeffmean(4),ventCoeffmean(5),ventCoeffmean
(6),ventCoeffmean(7),ventCoeffmean(8),ventCoeffmean(9),ventCoeffme
an(10));

```

```

C{61} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.16 WHERE gbauj <
1915;");',infiltrationRateMean(1));
C{63} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj BETWEEN
1915 AND 1945;");',infiltrationRateMean(2));
C{65} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj BETWEEN
1946 AND 1960;");',infiltrationRateMean(3));
C{67} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 1.16 WHERE gbauj BETWEEN
1961 AND 1970;");',infiltrationRateMean(4));
C{69} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.84 WHERE gbauj BETWEEN
1971 AND 1980;");',infiltrationRateMean(5));
C{71} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.43 WHERE gbauj BETWEEN
1981 AND 1990;");',infiltrationRateMean(6));
C{73} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.29 WHERE gbauj BETWEEN
1991 AND 2000;");',infiltrationRateMean(7));
C{75} = sprintf('ps = c.prepareStatement("UPDATE building SET
infiltration_rate = %d, roof_u_value = 0.21 WHERE gbauj >
2000;");',infiltrationRateMean(8));

name = sprintf('CitysimDatabaseLinkerXML%d.java',n);
[fileID, message] = fopen(name, 'wt+');
for k = 1:1:length(C)
    fprintf(fileID, '%s\n',C{k});
end
fclose(fileID);

command = sprintf('mingw32-make -f MakefileXML.mingw32
MAINCLASS=CitySimDatabaseLinkerXML%d run',n);
[status,cmdout] = system(command);
command = 'CitySimPro essai.xml';
[status,cmdout] = system(command);
command = sprintf('mingw32-make -f MakefileReader.mingw32
run');
[status,cmdout] = system(command);

% A test query
query='select 0.5*sum(((heat_supply-
heat_supply_sim)/heat_supply)^2)/count(*) as sme from calibration
where scenario_id_fk = 0'; % Gets all records
st=conn.createStatement();
rs=st.executeQuery(query);
rs.next();
smelm = str2double(char(rs.getString('sme')))
sme2m = 0.5*(sum([1 1 1 1 1 1 1 1 1 1]-natVentCoeffmean).^2)
+ sum([1 1 1 1 1 1 1 1 1 1]-ventCoeffmean).^2) +
1.5*sum(((infiltrationRateRef-
infiltrationRateMean)/infiltrationRateRef).^2))/3.5
Fmean(j) = q1*smelm + q2*sme2m;

```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

if Fmean(j) < Fc(j)
    Fc(j+1) = Fmean(j);
    Xc(j+1,:) = Xmean(j,:);
else
    Fc(j+1) = Fc(j);
    Xc(j+1,:) = Xc(j,:);
end
UB(j+1,:) = Xc(j+1,:)+Ro(j,:);
LB(j+1,:) = Xc(j+1,:)-Ro(j,:);
Ro(j+1,:) = (UB(j+1,:) - LB(j+1,:))/2;
Xmov(j) = (norm((Xc(j+1,:)-
Xc(j,:))./(2*Ro(j,:))))/sqrt(length(natVentCoeffUB));
if j>1
    dnn(j) = norm((Xbest(j,:)-Xbest(j-
1,:))./(2*Ro(j,:)))/sqrt(length(natVentCoeffUB));
    if F(j,i)<tolF && j<M
        countTol = countTol + 1;
        if countTol == 30;
            j = M+1;
        end
    elseif Fbest(j)>Fmean(j) && j<M
        countF = countF + 1;
        if countF == 30;
            j = M+1;
        end
    elseif dnn(j)<toldnn && j<M
        countD = countD + 1;
        if countD == 30;
            j = M+1;
        end
    end
end
if j<=M
    if Xmov(j) < 0.1
        movCount = movCount + 1;
        if movCount > 0
            S(j,:) = 1 - 0.2*exp(-3*dnn(j));
            Ro(j+1,:) = Ro(j,:).*S(j,:);
            UB(j+1,:) = Xc(j+1,:)+Ro(j+1,:);
            LB(j+1,:) = Xc(j+1,:)-Ro(j+1,:);
            movCount = 0;
        end
    end
end
end
j
volumehc = prod(S(:,1))
end

```





## Annex IV : PostgreSQL code used to create the table that contains all the data of the buildings.

```
-- Table: data.building
--DROP TABLE data.building;

CREATE TABLE data.building
(
  scenario_id_fk integer NOT NULL DEFAULT 0,
  height real,
  sre double precision DEFAULT 1000,
  gbaup numeric(4,0) DEFAULT 8017,
  grenp numeric(4,0),
  gbauj numeric(4,0) DEFAULT 1914,
  grenj numeric(4,0) DEFAULT 9999,
  gastw numeric(2,0) DEFAULT 6,
  status numeric(4,0) DEFAULT 1004,
  wall_av_u_value real,
  altitude real DEFAULT 373.6,
  address text DEFAULT 'Geneva_City'::text,
  footprint postgis.geometry,
  building_id_fk integer,
  description text,
  composite_id_fk integer DEFAULT 50,
  n_floors numeric(2,0) DEFAULT 4,
  window_g_value real NOT NULL DEFAULT 0.7,
  ground_k_value real NOT NULL DEFAULT 3,
  roof_u_value real NOT NULL DEFAULT 0.3,
  wall_shortwave_reflectance real NOT NULL DEFAULT 0.25,
  roof_shortwave_reflectance real NOT NULL DEFAULT 0.25,
  t_min_set_point real NOT NULL DEFAULT 21,
  t_max_set_point real NOT NULL DEFAULT 26,
  infiltration_rate real NOT NULL DEFAULT 0.4,
  frame_to_window_ratio real DEFAULT 0.3,
  shading_devices boolean NOT NULL DEFAULT false,
  adjacent_non_heated_spaces boolean NOT NULL DEFAULT false,
  fenestration_ratio_nesw real[] DEFAULT ARRAY[0.25, 0.0, 0.25,
(0)::numeric],
  fenestration_u_value_nesw real[] DEFAULT ARRAY[2.25, 0.0, 2.58,
(0)::numeric],
  fenestration_openable_ratio_nesw real[] DEFAULT ARRAY[2.25, 0.0,
2.58, 0.0],
  sia_category integer DEFAULT 1,
  floor_area real,
  volume real,
  egid integer,
  fuel_consumption numeric,
  indice real,
  purpose text,
  zone integer,
  buildingtype character varying,
  nofloor_statbl integer,
```

```

    noroom_statbl integer,
    nopeople integer,
    heatedarea double precision,
    typehousing text,
    occupancytype integer,
    blindslambda real,
    blindsirradiancecutoff real,
    minergiep_walltype integer,
    minergiep_floortype integer,
    minergiep_rooftype integer,
    minergiep_ventilation numeric,
    footprint_gis geometry NOT NULL,
    an_trans integer,
    window_u_value double precision,
    CONSTRAINT building_composite_id_fk_fkey FOREIGN KEY
(composite_id_fk)
    REFERENCES data.composite (composite_id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT building_scenario_id_fk_fkey FOREIGN KEY
(scenario_id_fk)
    REFERENCES data.scenario (scenario_id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT building_gastw_check CHECK (gastw >= 1::numeric AND
gastw <= 59::numeric),
    CONSTRAINT building_gbaup_check CHECK (gbaup >= 8011::numeric
AND gbaup <= 8023::numeric),
    CONSTRAINT enforce_dims_footprint CHECK (st_ndims(footprint_gis)
= 2),
    CONSTRAINT enforce_geotype_footprint CHECK
(geometrytype(footprint) = 'MULTIPOLYGON'::text OR footprint IS
NULL),
    CONSTRAINT enforce_srid_footprint CHECK (st_srid(footprint) =
2056)
)
WITH (
    OIDS=TRUE
);
ALTER TABLE data.building
    OWNER TO upadhyay;
GRANT ALL ON TABLE data.building TO upadhyay;
GRANT ALL ON TABLE data.building TO kaempf;

-- Trigger: check_orientation_before on data.building

-- DROP TRIGGER check_orientation_before ON data.building;

CREATE TRIGGER check_orientation_before
    BEFORE INSERT OR UPDATE
    ON data.building
    FOR EACH ROW
    EXECUTE PROCEDURE data.set_orientation();

--INSERTING DATA FROM SITG

```

```

INSERT INTO building (egid, footprint_gis) SELECT egid e,
ST_Multi(ST_Union(array_agg(footprint))) FROM sitg GROUP BY egid;
--15035
UPDATE building SET building_id_fk = q3.fk FROM (SELECT
s.building_id_fk fk, s.egid e FROM sitg s, (SELECT egid eg,
max(ST_Area(footprint)) area from sitg GROUP by egid) q1 WHERE
q1.eg = s.egid AND ST_Area(s.footprint) = q1.area) as q3 WHERE
q3.e = building.egid;

--FILLING IT WITH DATA FROM REGBL
WITH aux AS(select DISTINCT ON (egid) gbaup as a, gbauj as b,
gastw as c, concat(gdenamk, ' - ',strnamk1,', ',deinr) as e, gastw
as f, garea as g, egid as h, sum(ganzwhg) as i from regbl q1 where
exists( select 1 from sitg q2 where q1.egid = q2.egid)
group by gbaup, gbauj, gastw, concat(gdenamk, ' - ',strnamk1,',
',deinr), gastw, garea, egid)
UPDATE building SET
    gbaup=aux.a,
    gbauj=aux.b,
    gastw=aux.c,
    address=aux.e,
    n_floors=aux.f,
    floor_area=aux.g,
    egid=aux.h,
    nopeople=i
from aux
where building.egid = aux.h; --11050

--UPDATING THE NOPEOPLE VALUE ON THE BUILDING TABLE IN SUCH A WAY
THAT ALL THE VALUES OF THE GANZWGH COLUMN OF REGBL ARE SUMMED UP
(THERE ARE MANY ELEMENTS IN REGBL REFERRED TO THE SAME BUILDING
AND WITH THE SAME EGID) --- USED IN BUILDING!!!
--UPDATE building SET nopeople = a.b from (select egid,
sum(ganzwhg) as b from regbl GROUP BY regbl.egid) as a where
building.egid = a.egid; --11050

--ASSIGNING ONE ZONE TO EVERY BUILDING
UPDATE building SET zone = a.zone_id FROM (SELECT zone_id,
geom_gis FROM zone WHERE zone_id != 17) as a WHERE
ST_Contains(a.geom_gis,building.footprint_gis); --13785

--IN THE BUILDING TABLE THERE ARE MANY ELEMENTS WITH THE SAME
EGID, SOME OF THEM HVE SMALL FOOTPRINTS, WE NEED TO ESTIMATE THE
NUMBER OF PEOPLE FOR EACH OF THESE ELEMENTS IN A PROPORTION TO
THEIR RESPECTIVE FOOTPRINT AREAS
--UPDATE building SET nopeople =
round(nopeople*ST_Area(footprint_gis)/floor_area) FROM (select
distinct(egid) as c, count(*) as b from building group by egid) as
a where a.b >1 and a.c=building.egid; --3016

--UPDATING SOME OTHER VALUES OF THE BUILDING TABLE
UPDATE building SET height = a.hauteur, purpose = a.destinat from
(select hauteur, destinat, building_id_fk from sitg) as a where

```

```

a.building_id_fk=building.building_id_fk; --15035 ---USED IN
BUILDING

--UPDATING HEATED AREA, VOLUME COLUMN, FLOOR_AREA COLUMN AND SOME
OTHER COLUMNS WITH STANDARD VALUES THAT WE'LL CHANGE LATER ON
UPDATE building SET floor_area = ST_Area(footprint_gis),
heatedarea = floor_area*n_floors, volume = height*floor_area,
minergiep_walltype = 68, minergiep_floortype = 70,
minergiep_rooftype = 69, composite_id_fk = 56, blindslambda =
0.017, minergiep_ventilation = 0.33, blindsirradiancecutoff = 300,
window_g_value = 0.47, roof_u_value = 0.43,
wall_shortwave_reflectance = 0.25, roof_shortwave_reflectance =
0.25; --15035

--UPDATING THE NUMBER OF FLOORS OF THE BUILDINGS: WE GIVE PRIORITY
TO THE DATA THAT COMES FROM REGBL, IF THE NUMBER OF FLOOR IS NOT
THERE, WE USE SITG
UPDATE building SET gastw = a.gastw, n_floors = a.gastw from regbl
a where a.egid = building.egid; --11050 inserted, but many are
null
UPDATE building SET gastw = a.niv_hs, n_floors = a.niv_hs from
sitg a where a.building_id_fk = building.building_id_fk and not
exists(select 1 from regbl b where b.egid = building.egid); --
3985 inserted, but many are null too
UPDATE building SET gastw = a.niv_hs, n_floors = a.niv_hs from
sitg a where a.building_id_fk = building.building_id_fk and
exists(select 1 from regbl b where b.egid = building.egid AND
b.gastw IS NULL); --117 , some might be null

--UPDATING THE YEAR OF CONSTRUCTION
UPDATE building SET gbauj = a.gbauj from (select gbauj, egid from
regbl) as a where a.egid = building.egid and a.gbauj is not null;
--ONLY 1836 not null inserted

--UPDATING THE YEAR OF CONSTRUCTION FROM THE COLUMN GBAUP OF
REGBL
UPDATE building SET gbauj = 1900 WHERE gbaup = 8011 AND gbauj IS
NULL; --2628
UPDATE building SET gbauj = 1934 WHERE gbaup = 8012 AND gbauj IS
NULL; -- 1537
UPDATE building SET gbauj = 1952 WHERE gbaup = 8013 AND gbauj IS
NULL; -- 1410
UPDATE building SET gbauj = 1966 WHERE gbaup = 8014 AND gbauj IS
NULL; -- 1068
UPDATE building SET gbauj = 1976 WHERE gbaup = 8015 AND gbauj IS
NULL; -- 714
UPDATE building SET gbauj = 1983 WHERE gbaup = 8016 AND gbauj IS
NULL; -- 292
UPDATE building SET gbauj = 1988 WHERE gbaup = 8017 AND gbauj IS
NULL; -- 571
UPDATE building SET gbauj = 1993 WHERE gbaup = 8018 AND gbauj IS
NULL; -- 390
UPDATE building SET gbauj = 1998 WHERE gbaup = 8019 AND gbauj IS
NULL; -- 438

```

```

UPDATE building SET gbauj = 2003 WHERE gbaup = 8020 AND gbauj IS
NULL; -- 0
UPDATE building SET gbauj = 2008 WHERE gbaup = 8021 AND gbauj IS
NULL; -- 0
UPDATE building SET gbauj = 2013 WHERE gbaup = 8022 AND gbauj IS
NULL; -- 0
UPDATE building SET gbauj = 2016 WHERE gbaup = 8023 AND gbauj IS
NULL; -- 0

--UPDATING FROM SITG
UPDATE building SET gbauj = to_number(a.an_constr,'9999') from
(select an_constr, building_id_fk from sitg) as a where
a.building_id_fk=building.building_id_fk and building.gbauj is
null and a.an_constr is not null; --181 not null inserted

--UPDATING THE YEAR OF CONSTRUCTION FROM THE EPOQUE OF
CONSTRUCTION
UPDATE building SET gbauj = a.b FROM (select
round((to_number(substring(ep_constr,1,4),'9999') +
to_number(substring(ep_constr,6,8),'9999'))/2) b, building_id_fk
from sitg where substring(ep_constr,1,4) not like 'Avan' and
substring(ep_constr,1,4) not like 'EAPA' and
substring(ep_constr,5,6) like '-%') as a WHERE a.building_id_fk =
building.building_id_fk AND building.gbauj is null; --26 rows
updated

UPDATE building SET gbauj = a.b FROM (select
to_number(substring(ep_constr,6,8),'9999') b, building_id_fk from
sitg where substring(ep_constr,1,4) like 'Avan') as a WHERE
a.building_id_fk = building.building_id_fk AND building.gbauj is
null; --8 rows updated

UPDATE building SET gbauj = a.b FROM (select
to_number(substring(ep_constr,1,4),'9999') b, building_id_fk from
sitg where substring(ep_constr,5,6) not like '%-' and
substring(ep_constr,1,4) not like 'Avan' and
substring(ep_constr,1,4) not like 'EAPA') as a WHERE
a.building_id_fk = building.building_id_fk AND building.gbauj is
null; --0 rows inserted

--some buildings might have different building_id_fk but the same
EGID, if the date of construction is not in one of them we can
update it with the date of the other element of the same EGID
--UPDATE building SET gbauj = a.b FROM (select gbauj b, egid c
from building where gbauj is not null) as a where a.c =
building.egid and building.gbauj is null; --0 rows inserted

-- UPDATE THE REST OF THE DATES OF CONSTRUCTION BY USING A
MONTECARLO APPROXIMATION TO FIND THE CORRESPONDING EPOQUES (1MIN
TO RUN)
CREATE OR REPLACE FUNCTION updateUsingMontecarlo()
RETURNS void AS
$BODY$

```

```

DECLARE
i integer;
BEGIN
SELECT INTO i max(query.counter) FROM (SELECT distinct(zone),
count(*) counter FROM building WHERE gbauj is null group by zone)
as query;
FOR i IN 1..i LOOP
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.013740458
then round(1870+(a.const-0)*(1914-1870)/(0.013740458-0)) WHEN
a.const between 0.013740458 and 0.410687023 then
round(1915+(a.const-0.013740458)*(1935-1915)/(0.410687023-
0.013740458)) WHEN a.const between 0.410687023 and 0.422900763
then round(1935+(a.const-0.410687023)*(1950-1935)/(0.422900763-
0.410687023)) WHEN a.const between 0.422900763 and 0.667175573
then round(1950+(a.const-0.422900763)*(1970-1950)/(0.667175573-
0.422900763)) WHEN a.const between 0.667175573 and 0.858015267
then round(1970+(a.const-0.667175573)*(1990-1970)/(0.858015267-
0.667175573)) WHEN a.const between 0.858015267 and 0.93740458
then round(1990+(a.const-0.858015267)*(2000-1990)/(0.93740458-
0.858015267)) WHEN a.const between 0.93740458 and 1 then
round(2000+(a.const-0.93740458)*(2015-2000)/(1-0.93740458)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 1 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 1 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.329059829
then round(1870+(a.const-0)*(1914-1870)/(0.329059829-0)) WHEN
a.const between 0.329059829 and 0.329059829 then
round(1915+(a.const-0.329059829)*(1935-1915)/(0.329059829-
0.329059829)) WHEN a.const between 0.329059829 and 0.341880342
then round(1935+(a.const-0.329059829)*(1950-1935)/(0.341880342-
0.329059829)) WHEN a.const between 0.341880342 and 0.618233618
then round(1950+(a.const-0.341880342)*(1970-1950)/(0.618233618-
0.341880342)) WHEN a.const between 0.618233618 and 0.762108262
then round(1970+(a.const-0.618233618)*(1990-1970)/(0.762108262-
0.618233618)) WHEN a.const between 0.762108262 and 0.867521368
then round(1990+(a.const-0.762108262)*(2000-1990)/(0.867521368-
0.762108262)) WHEN a.const between 0.867521368 and 1 then
round(2000+(a.const-0.867521368)*(2015-2000)/(1-0.867521368)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 2 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 2 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.180722892
then round(1870+(a.const-0)*(1914-1870)/(0.180722892-0)) WHEN
a.const between 0.180722892 and 0.180722892 then
round(1915+(a.const-0.180722892)*(1935-1915)/(0.180722892-
0.180722892)) WHEN a.const between 0.180722892 and 0.192771084
then round(1935+(a.const-0.180722892)*(1950-1935)/(0.192771084-

```



```

0.180722892)) WHEN a.const between 0.192771084 and 0.277108434
then round(1950+(a.const-0.192771084)*(1970-1950)/(0.277108434-
0.192771084)) WHEN a.const between 0.277108434 and 0.530120482
then round(1970+(a.const-0.277108434)*(1990-1970)/(0.530120482-
0.277108434)) WHEN a.const between 0.530120482 and 0.698795181
then round(1990+(a.const-0.530120482)*(2000-1990)/(0.698795181-
0.530120482)) WHEN a.const between 0.698795181 and 1 then
round(2000+(a.const-0.698795181)*(2015-2000)/(1-0.698795181)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 3 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 3 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.028169014
then round(1870+(a.const-0)*(1914-1870)/(0.028169014-0)) WHEN
a.const between 0.028169014 and 0.366197183 then
round(1915+(a.const-0.028169014)*(1935-1915)/(0.366197183-
0.028169014)) WHEN a.const between 0.366197183 and 0.387323944
then round(1935+(a.const-0.366197183)*(1950-1935)/(0.387323944-
0.366197183)) WHEN a.const between 0.387323944 and 0.64084507
then round(1950+(a.const-0.387323944)*(1970-1950)/(0.64084507-
0.387323944)) WHEN a.const between 0.64084507 and 0.697183099
then round(1970+(a.const-0.64084507)*(1990-1970)/(0.697183099-
0.64084507)) WHEN a.const between 0.697183099 and 0.73943662
then round(1990+(a.const-0.697183099)*(2000-1990)/(0.73943662-
0.697183099)) WHEN a.const between 0.73943662 and 1 then
round(2000+(a.const-0.73943662)*(2015-2000)/(1-0.73943662)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 4 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 4 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0
then round(1870+(a.const-0)*(1914-1870)/(0-0)) WHEN
a.const between 0 and 0.342105263 then
round(1915+(a.const-0)*(1935-1915)/(0.342105263-0))
WHEN a.const between 0.342105263 and 0.35 then
round(1935+(a.const-0.342105263)*(1950-1935)/(0.35-0.342105263))
WHEN a.const between 0.35 and 0.786842105 then
round(1950+(a.const-0.35)*(1970-1950)/(0.786842105-0.35))
WHEN a.const between 0.786842105 and 0.821052632 then
round(1970+(a.const-0.786842105)*(1990-1970)/(0.821052632-
0.786842105)) WHEN a.const between 0.821052632 and 0.847368421
then round(1990+(a.const-0.821052632)*(2000-1990)/(0.847368421-
0.821052632)) WHEN a.const between 0.847368421 and 1 then
round(2000+(a.const-0.847368421)*(2015-2000)/(1-0.847368421)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 5 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 5 and y.z = building.building_id_fk;

```

```

WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.303341902
then round(1870+(a.const-0)*(1914-1870)/(0.303341902-0)) WHEN
a.const between 0.303341902 and 0.303341902 then
round(1915+(a.const-0.303341902)*(1935-1915)/(0.303341902-
0.303341902)) WHEN a.const between 0.303341902 and 0.368894602
then round(1935+(a.const-0.303341902)*(1950-1935)/(0.368894602-
0.303341902)) WHEN a.const between 0.368894602 and 0.637532134
then round(1950+(a.const-0.368894602)*(1970-1950)/(0.637532134-
0.368894602)) WHEN a.const between 0.637532134 and 0.732647815
then round(1970+(a.const-0.637532134)*(1990-1970)/(0.732647815-
0.637532134)) WHEN a.const between 0.732647815 and 0.894601542
then round(1990+(a.const-0.732647815)*(2000-1990)/(0.894601542-
0.732647815)) WHEN a.const between 0.894601542 and 1 then
round(2000+(a.const-0.894601542)*(2015-2000)/(1-0.894601542)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 6 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 6 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.001945525
then round(1870+(a.const-0)*(1914-1870)/(0.001945525-0)) WHEN
a.const between 0.001945525 and 0.424124514 then
round(1915+(a.const-0.001945525)*(1935-1915)/(0.424124514-
0.001945525)) WHEN a.const between 0.424124514 and 0.459143969
then round(1935+(a.const-0.424124514)*(1950-1935)/(0.459143969-
0.424124514)) WHEN a.const between 0.459143969 and 0.73151751
then round(1950+(a.const-0.459143969)*(1970-1950)/(0.73151751-
0.459143969)) WHEN a.const between 0.73151751 and 0.770428016
then round(1970+(a.const-0.73151751)*(1990-1970)/(0.770428016-
0.73151751)) WHEN a.const between 0.770428016 and 0.869649805
then round(1990+(a.const-0.770428016)*(2000-1990)/(0.869649805-
0.770428016)) WHEN a.const between 0.869649805 and 1 then
round(2000+(a.const-0.869649805)*(2015-2000)/(1-0.869649805)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 7 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 7 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.031746032
then round(1870+(a.const-0)*(1914-1870)/(0.031746032-0)) WHEN
a.const between 0.031746032 and 0.650793651 then
round(1915+(a.const-0.031746032)*(1935-1915)/(0.650793651-
0.031746032)) WHEN a.const between 0.650793651 and 0.66031746
then round(1935+(a.const-0.650793651)*(1950-1935)/(0.66031746-
0.650793651)) WHEN a.const between 0.66031746 and 0.777777778
then round(1950+(a.const-0.66031746)*(1970-1950)/(0.777777778-
0.66031746)) WHEN a.const between 0.777777778 and 0.911111111
then round(1970+(a.const-0.777777778)*(1990-1970)/(0.911111111-
0.777777778)) WHEN a.const between 0.911111111 and 0.93968254
then round(1990+(a.const-0.911111111)*(2000-1990)/(0.93968254-
0.911111111)) WHEN a.const between 0.93968254 and 1 then

```

```

round(2000+(a.const-0.93968254)*(2015-2000)/(1-0.93968254)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 8 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 8 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.004846527
then round(1870+(a.const-0)*(1914-1870)/(0.004846527-0)) WHEN
a.const between 0.004846527 and 0.484652666 then
round(1915+(a.const-0.004846527)*(1935-1915)/(0.484652666-
0.004846527)) WHEN a.const between 0.484652666 and 0.495961228
then round(1935+(a.const-0.484652666)*(1950-1935)/(0.495961228-
0.484652666)) WHEN a.const between 0.495961228 and 0.684975767
then round(1950+(a.const-0.495961228)*(1970-1950)/(0.684975767-
0.495961228)) WHEN a.const between 0.684975767 and 0.815831987
then round(1970+(a.const-0.684975767)*(1990-1970)/(0.815831987-
0.684975767)) WHEN a.const between 0.815831987 and 0.89822294
then round(1990+(a.const-0.815831987)*(2000-1990)/(0.89822294-
0.815831987)) WHEN a.const between 0.89822294 and 1 then
round(2000+(a.const-0.89822294)*(2015-2000)/(1-0.89822294)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 9 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 9 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.003546099
then round(1870+(a.const-0)*(1914-1870)/(0.003546099-0)) WHEN
a.const between 0.003546099 and 0.5 then
round(1915+(a.const-0.003546099)*(1935-1915)/(0.5-0.003546099))
WHEN a.const between 0.5 and 0.514184397 then
round(1935+(a.const-0.5)*(1950-1935)/(0.514184397-0.5))
WHEN a.const between 0.514184397 and 0.687943262 then
round(1950+(a.const-0.514184397)*(1970-1950)/(0.687943262-
0.514184397)) WHEN a.const between 0.687943262 and 0.84929078
then round(1970+(a.const-0.687943262)*(1990-1970)/(0.84929078-
0.687943262)) WHEN a.const between 0.84929078 and 0.955673759
then round(1990+(a.const-0.84929078)*(2000-1990)/(0.955673759-
0.84929078)) WHEN a.const between 0.955673759 and 1 then
round(2000+(a.const-0.955673759)*(2015-2000)/(1-0.955673759)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 10 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 10 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.004545455
then round(1870+(a.const-0)*(1914-1870)/(0.004545455-0)) WHEN
a.const between 0.004545455 and 0.457575758 then
round(1915+(a.const-0.004545455)*(1935-1915)/(0.457575758-
0.004545455)) WHEN a.const between 0.457575758 and 0.471212121
then round(1935+(a.const-0.457575758)*(1950-1935)/(0.471212121-
0.457575758)) WHEN a.const between 0.471212121 and 0.686363636

```

```

then round(1950+(a.const-0.471212121)*(1970-1950)/(0.686363636-
0.471212121)) WHEN a.const between 0.686363636 and 0.856060606
then round(1970+(a.const-0.686363636)*(1990-1970)/(0.856060606-
0.686363636)) WHEN a.const between 0.856060606 and 0.972727273
then round(1990+(a.const-0.856060606)*(2000-1990)/(0.972727273-
0.856060606)) WHEN a.const between 0.972727273 and 1 then
round(2000+(a.const-0.972727273)*(2015-2000)/(1-0.972727273)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 11 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 11 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.005181347
then round(1870+(a.const-0)*(1914-1870)/(0.005181347-0)) WHEN
a.const between 0.005181347 and 0.371761658 then
round(1915+(a.const-0.005181347)*(1935-1915)/(0.371761658-
0.005181347)) WHEN a.const between 0.371761658 and 0.378238342
then round(1935+(a.const-0.371761658)*(1950-1935)/(0.378238342-
0.371761658)) WHEN a.const between 0.378238342 and 0.637305699
then round(1950+(a.const-0.378238342)*(1970-1950)/(0.637305699-
0.378238342)) WHEN a.const between 0.637305699 and 0.813471503
then round(1970+(a.const-0.637305699)*(1990-1970)/(0.813471503-
0.637305699)) WHEN a.const between 0.813471503 and 0.888601036
then round(1990+(a.const-0.813471503)*(2000-1990)/(0.888601036-
0.813471503)) WHEN a.const between 0.888601036 and 1 then
round(2000+(a.const-0.888601036)*(2015-2000)/(1-0.888601036)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 12 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 12 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.021220159
then round(1870+(a.const-0)*(1914-1870)/(0.021220159-0)) WHEN
a.const between 0.021220159 and 0.786472149 then
round(1915+(a.const-0.021220159)*(1935-1915)/(0.786472149-
0.021220159)) WHEN a.const between 0.786472149 and 0.789124668
then round(1935+(a.const-0.786472149)*(1950-1935)/(0.789124668-
0.786472149)) WHEN a.const between 0.789124668 and 0.879310345
then round(1950+(a.const-0.789124668)*(1970-1950)/(0.879310345-
0.789124668)) WHEN a.const between 0.879310345 and 0.940318302
then round(1970+(a.const-0.879310345)*(1990-1970)/(0.940318302-
0.879310345)) WHEN a.const between 0.940318302 and 0.966843501
then round(1990+(a.const-0.940318302)*(2000-1990)/(0.966843501-
0.940318302)) WHEN a.const between 0.966843501 and 1 then
round(2000+(a.const-0.966843501)*(2015-2000)/(1-0.966843501)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 13 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 13 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.015228426

```

```

then round(1870+(a.const-0)*(1914-1870)/(0.015228426-0)) WHEN
a.const between 0.015228426 and 0.455837563 then
round(1915+(a.const-0.015228426)*(1935-1915)/(0.455837563-
0.015228426)) WHEN a.const between 0.455837563 and 0.469035533
then round(1935+(a.const-0.455837563)*(1950-1935)/(0.469035533-
0.455837563)) WHEN a.const between 0.469035533 and 0.674111675
then round(1950+(a.const-0.469035533)*(1970-1950)/(0.674111675-
0.469035533)) WHEN a.const between 0.674111675 and 0.864974619
then round(1970+(a.const-0.674111675)*(1990-1970)/(0.864974619-
0.674111675)) WHEN a.const between 0.864974619 and 0.964467005
then round(1990+(a.const-0.864974619)*(2000-1990)/(0.964467005-
0.864974619)) WHEN a.const between 0.964467005 and 1 then
round(2000+(a.const-0.964467005)*(2015-2000)/(1-0.964467005)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 14 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 14 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.004731861
then round(1870+(a.const-0)*(1914-1870)/(0.004731861-0)) WHEN
a.const between 0.004731861 and 0.345425868 then
round(1915+(a.const-0.004731861)*(1935-1915)/(0.345425868-
0.004731861)) WHEN a.const between 0.345425868 and 0.361198738
then round(1935+(a.const-0.345425868)*(1950-1935)/(0.361198738-
0.345425868)) WHEN a.const between 0.361198738 and 0.594637224
then round(1950+(a.const-0.361198738)*(1970-1950)/(0.594637224-
0.361198738)) WHEN a.const between 0.594637224 and 0.840694006
then round(1970+(a.const-0.594637224)*(1990-1970)/(0.840694006-
0.594637224)) WHEN a.const between 0.840694006 and 0.933753943
then round(1990+(a.const-0.840694006)*(2000-1990)/(0.933753943-
0.840694006)) WHEN a.const between 0.933753943 and 1 then
round(2000+(a.const-0.933753943)*(2015-2000)/(1-0.933753943)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone = 15 and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 15 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.003521127
then round(1870+(a.const-0)*(1914-1870)/(0.003521127-0)) WHEN
a.const between 0.003521127 and 0.366197183 then
round(1915+(a.const-0.003521127)*(1935-1915)/(0.366197183-
0.003521127)) WHEN a.const between 0.366197183 and 0.36971831
then round(1935+(a.const-0.366197183)*(1950-1935)/(0.36971831-
0.366197183)) WHEN a.const between 0.36971831 and 0.647887324
then round(1950+(a.const-0.36971831)*(1970-1950)/(0.647887324-
0.36971831)) WHEN a.const between 0.647887324 and 0.86971831
then round(1970+(a.const-0.647887324)*(1990-1970)/(0.86971831-
0.647887324)) WHEN a.const between 0.86971831 and 0.926056338
then round(1990+(a.const-0.86971831)*(2000-1990)/(0.926056338-
0.86971831)) WHEN a.const between 0.926056338 and 1 then
round(2000+(a.const-0.926056338)*(2015-2000)/(1-0.926056338)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building

```

```

where zone = 16      and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone = 16 and y.z = building.building_id_fk;
WITH a as (SELECT random() as const) UPDATE building SET gbauj =
b.date from (SELECT CASE WHEN a.const between 0 and 0.16016368
then round(1870+(a.const-0)*(1914-1870)/(0.16016368-0))    WHEN
a.const between 0.16016368 and 0.336006195 then
round(1915+(a.const-0.16016368)*(1935-1915)/(0.336006195-
0.16016368))    WHEN a.const between 0.336006195 and 0.360422998
then round(1935+(a.const-0.336006195)*(1950-1935)/(0.360422998-
0.336006195))    WHEN a.const between 0.360422998 and 0.624572575
then round(1950+(a.const-0.360422998)*(1970-1950)/(0.624572575-
0.360422998))    WHEN a.const between 0.624572575 and 0.801292098
then round(1970+(a.const-0.624572575)*(1990-1970)/(0.801292098-
0.624572575))    WHEN a.const between 0.801292098 and 0.905483298
then round(1990+(a.const-0.801292098)*(2000-1990)/(0.905483298-
0.801292098))    WHEN a.const between 0.905483298 and 1 then
round(2000+(a.const-0.905483298)*(2015-2000)/(1-0.905483298)) ELSE
0 END AS date FROM a) as b, (SELECT building_id_fk z from building
where zone is null and gbauj is null AND NOT EXISTS (SELECT 1
FROM building as new WHERE gbauj is not null and new.egid =
building.egid) limit 1) as y WHERE building.gbauj is null and
building.zone is null and y.z = building.building_id_fk;
END LOOP;
RETURN;
END;
$BODY$
LANGUAGE plpgsql;
SELECT updateUsingMontecarlo();
DROP FUNCTION IF EXISTS updateUsingMontecarlo();

```

```

--some buildings might have different building_id_fk but the same
EGID, is the date of construction is not in one of them we can
update it with the date of the other element of the same EGID
--UPDATE building SET gbauj = a.b FROM (select gbauj b, egid c
from building where gbauj is not null) as a where a.c =
building.egid and building.gbauj is null; --701 rows inserted

```

```

--NOW WE CAN UPDATE THE HEIGHTS AND FLOORS BASED ON THESE VALUES
UPDATE building SET height = gastw*4.4497296701262 WHERE gbauj
between 1700 and 1914 and height is null and gastw is not null; --
22
UPDATE building SET height = gastw*4.2361992682332 WHERE gbauj
between 1915 and 1934 and height is null and gastw is not null; --
67
UPDATE building SET height = gastw*3.81803455569687 WHERE gbauj
between 1935 and 1949 and height is null and gastw is not null; --
4
UPDATE building SET height = gastw*3.81085881483094 WHERE gbauj
between 1950 and 1969 and height is null and gastw is not null; --
46

```

```

UPDATE building SET height = gastw*3.82907119376942 WHERE gbauj
between 1970 and 1989 and height is null and gastw is not null; --
40
UPDATE building SET height = gastw*3.44797821489881 WHERE gbauj
between 1990 and 1999 and height is null and gastw is not null; --
29
UPDATE building SET height = gastw*3.36675098655043 WHERE gbauj
between 2000 and 2016 and height is null and gastw is not null; --
180

--NOW WE UPDATE THE NUMBER OF FLOORS
UPDATE building SET gastw = trunc(height/4.4497296701262+1),
n_floors = trunc(height/4.4497296701262+1) WHERE gbauj between
1700 and 1914 and gastw is null and height is not null; --2630
UPDATE building SET gastw = trunc(height/4.2361992682332+1),
n_floors = trunc(height/4.2361992682332+1) WHERE gbauj between
1915 and 1934 and gastw is null and height is not null; --28
UPDATE building SET gastw = trunc(height/3.81803455569687+1),
n_floors = trunc(height/3.81803455569687+1) WHERE gbauj between
1935 and 1949 and gastw is null and height is not null; --62
UPDATE building SET gastw = trunc(height/3.81085881483094+1) WHERE gbauj between
1950 and 1969 and gastw is null and height is not null; --20
UPDATE building SET gastw = trunc(height/3.82907119376942+1),
n_floors = trunc(height/3.82907119376942+1) WHERE gbauj between
1970 and 1989 and gastw is null and height is not null; --16
UPDATE building SET gastw = trunc(height/3.44797821489881+1),
n_floors = trunc(height/3.44797821489881+1) WHERE gbauj between
1990 and 1999 and gastw is null and height is not null; --7
UPDATE building SET gastw = trunc(height/3.36675098655043+1),
n_floors = trunc(height/3.36675098655043+1) WHERE gbauj between
2000 and 2016 and gastw is null and height is not null; --12

--some buildings might have different building_id_fk but the same
EGID, if the height and floors is not in one of them we can update
it with the date of the other element of the same EGID (only for
those buildidns that have a very small footprint area)
--UPDATE building SET gastw = a.b, n_floors = a.b, height = a.d
FROM (select gastw b, egid c, height d from building where gastw
is not null and height is not null) as a where a.c = building.egid
and building.height is null and building.gastw is null; --0 rows
inserted

UPDATE building SET height = 3 WHERE ST_Area(footprint_gis) < 10
AND height is NULL;
UPDATE building SET height = 3.5 WHERE ST_Area(footprint_gis) < 50
AND height is NULL;

--UPDATING THE HEIGHTS
CREATE OR REPLACE FUNCTION updateHeightsWithLoop()
RETURNS void AS
$BODY$
DECLARE
i int;

```

```

BEGIN
FOR i IN 1..16 LOOP
UPDATE building SET height = a.h, gastw = trunc(a.h/a.rate+1),
n_floors = trunc(a.h/rate+1) FROM (
SELECT gbauj date, count(*), sum(height/gastw)/count(*) rate,
sum(height)/count(*) h FROM(
SELECT CASE
WHEN gbauj between 1700 and 1914 then '1700-1914' WHEN gbauj
between 1915 and 1934 then '1915-1934' WHEN gbauj between 1935 and
1949 then '1935-1949' WHEN gbauj between 1950 and 1969 then '1950-
1969'
WHEN gbauj between 1970 and 1989 then '1970-1989' WHEN gbauj
between 1990 and 1999 then '1990-1999' WHEN gbauj between 2000 and
2016 then '2000-2016'
ELSE 'NULL' END AS gbauj, gastw, height, zone
FROM data.building) gbauj
WHERE gbauj is not null AND height is not null and gastw is not
null and zone = i
GROUP BY gbauj ORDER BY gbauj) as a
WHERE building.gbauj between
to_number(substring(a.date,1,5), '9999') and
to_number(substring(a.date,6,8), '9999')
AND building.zone = i
AND building.height is null
AND building.gastw is null;
END LOOP;
RETURN;
END;
$BODY$
LANGUAGE plpgsql;
SELECT updateHeightsWithLoop();
DROP FUNCTION IF EXISTS updateHeightsWithLoop();

UPDATE building SET height = a.h, gastw = trunc(a.h/a.rate+1),
n_floors = trunc(a.h/rate+1) FROM (
SELECT gbauj date, count(*), sum(height/gastw)/count(*) rate,
sum(height)/count(*) h FROM(
SELECT CASE
WHEN gbauj between 1700 and 1914 then '1700-1914' WHEN gbauj
between 1915 and 1934 then '1915-1934' WHEN gbauj between 1935 and
1949 then '1935-1949' WHEN gbauj between 1950 and 1969 then '1950-
1969'
WHEN gbauj between 1970 and 1989 then '1970-1989' WHEN gbauj
between 1990 and 1999 then '1990-1999' WHEN gbauj between 2000 and
2016 then '2000-2016'
ELSE 'NULL' END AS gbauj, gastw, height, zone
FROM data.building) gbauj
WHERE gbauj is not null AND height is not null and gastw is not
null and zone =16 or zone = 15 or zone = 2 or zone = 6 or zone is
null
GROUP BY gbauj ORDER BY gbauj) as a
WHERE building.gbauj between
to_number(substring(a.date,1,5), '9999') and
to_number(substring(a.date,6,8), '9999')

```



```

AND building.zone IS NULL
AND building.height is null
AND building.gastw is null;

--UPDATING THE NUMBER OF FLOORS
CREATE OR REPLACE FUNCTION updateFloorsWithLoop()
RETURNS void AS
$BODY$
BEGIN
FOR i IN 1..16 LOOP
UPDATE building SET gastw = trunc(a.h/a.rate+1), n_floors =
trunc(a.h/rate+1) FROM (
SELECT gbauj date, count(*), sum(height/gastw)/count(*) rate,
sum(height)/count(*) h FROM(
SELECT CASE
WHEN gbauj between 1700 and 1914 then '1700-1914' WHEN gbauj
between 1915 and 1934 then '1915-1934' WHEN gbauj between 1935 and
1949 then '1935-1949' WHEN gbauj between 1950 and 1969 then '1950-
1969'
WHEN gbauj between 1970 and 1989 then '1970-1989' WHEN gbauj
between 1990 and 1999 then '1990-1999' WHEN gbauj between 2000 and
2016 then '2000-2016'
ELSE 'NULL' END AS gbauj, gastw, height, zone
FROM data.building) gbauj
WHERE gbauj is not null AND height is not null and gastw is not
null and zone = i
GROUP BY gbauj ORDER BY gbauj) as a
WHERE building.gbauj between
to_number(substring(a.date,1,5), '9999') and
to_number(substring(a.date,6,8), '9999')
AND building.zone = i
AND gastw is null
AND n_floors is null;
END LOOP;
RETURN;
END;
$BODY$
LANGUAGE plpgsql;
SELECT updateFloorsWithLoop();
DROP FUNCTION IF EXISTS updateFloorsWithLoop();

UPDATE building SET gastw = trunc(a.h/a.rate+1), n_floors =
trunc(a.h/rate+1) FROM (
SELECT gbauj date, count(*), sum(height/gastw)/count(*) rate,
sum(height)/count(*) h FROM(
SELECT CASE
WHEN gbauj between 1700 and 1914 then '1700-1914' WHEN gbauj
between 1915 and 1934 then '1915-1934' WHEN gbauj between 1935 and
1949 then '1935-1949' WHEN gbauj between 1950 and 1969 then '1950-
1969'
WHEN gbauj between 1970 and 1989 then '1970-1989' WHEN gbauj
between 1990 and 1999 then '1990-1999' WHEN gbauj between 2000 and
2016 then '2000-2016'
ELSE 'NULL' END AS gbauj, gastw, height, zone

```

```

FROM data.building) gbauj
WHERE gbauj is not null AND height is not null and gastw is not
null and zone =16 or zone = 15 or zone = 2 or zone = 6 or zone is
null
GROUP BY gbauj ORDER BY gbauj) as a
WHERE building.gbauj between
to_number(substring(a.date,1,5), '9999') and
to_number(substring(a.date,6,8), '9999')
AND building.zone is null
AND gastw is null
AND n_floors is null;

--REVERSING ALL THE GEOMETRIES IN SUCH A WAY THAT THE WALLS POINT
IN THE CORRECT DIRECTIONS
UPDATE data.building SET footprint_gis =
ST_Reverse(footprint_gis);

--CHECKING THAT THERE ARE NO BUILDINGS WITH FLOORS OF LESS THAN
2.5m HIGH (OPTIONAL)
--UPDATE building SET gastw = trunc(height/2.5), n_floors =
height/2.5 WHERE height/gastw < 2.5 AND gastw > 1 AND height >=
2.5; --560

UPDATE building SET composite_id_fk = a.composite_id FROM (SELECT
composite_id, period FROM composite) AS a WHERE gbauj BETWEEN
to_number(SUBSTRING(period,1,5), '9999') AND
to_number(SUBSTRING(period,6,8), '9999'); --12932
UPDATE building SET composite_id_fk = 105 WHERE gbauj < 1900; --64
UPDATE building SET composite_id_fk = 111 WHERE gbauj > 2010; --
141
UPDATE building SET wall_av_u_value = a.u_value FROM (SELECT
composite_id, u_value FROM composite) AS a WHERE a.composite_id =
composite_id_fk;

--UPDATING YEAR OF TRANSFORMATION
UPDATE building SET an_trans = a.b FROM (SELECT an_trans b,
building_id_fk fk FROM sitg) as a WHERE building_id_fk = a.fk;

--UPDATING VALUES FOR TRANSFORMATION
UPDATE building SET composite_id_fk = a.composite_id FROM (SELECT
composite_id, period FROM composite) AS a WHERE an_trans IS NOT
NULL AND an_trans BETWEEN to_number(SUBSTRING(period,1,5), '9999')
AND to_number(SUBSTRING(period,6,8), '9999'); --263
UPDATE building SET composite_id_fk = 112 WHERE an_trans IS NOT
NULL AND an_trans < 1900; --0
UPDATE building SET composite_id_fk = 115 WHERE an_trans IS NOT
NULL AND an_trans > 2010; --51

--UPDATING THE INFILTRATION RATES (Data from Diane's thesis:
original version)
UPDATE building SET infiltration_rate = 1.0779, roof_u_value =
0.16 WHERE gbauj < 1915; --6542
UPDATE building SET infiltration_rate = 1.1787, roof_u_value =
1.16 WHERE gbauj BETWEEN 1915 AND 1945; --6542

```

```

UPDATE building SET infiltration_rate = 1.2239, roof_u_value =
1.16 WHERE gbauj BETWEEN 1946 AND 1960; --2131
UPDATE building SET infiltration_rate = 1.300, roof_u_value = 1.16
WHERE gbauj BETWEEN 1961 AND 1970; --1104
UPDATE building SET infiltration_rate = 1.19121, roof_u_value =
0.84 WHERE gbauj BETWEEN 1971 AND 1980; --759
UPDATE building SET infiltration_rate = 1.09325, roof_u_value =
0.43 WHERE gbauj BETWEEN 1981 AND 1990; --893
UPDATE building SET infiltration_rate = 0.9200, roof_u_value =
0.29 WHERE gbauj BETWEEN 1991 AND 2000; --1061
UPDATE building SET infiltration_rate = 0.59897, roof_u_value =
0.21 WHERE gbauj > 2000; --200

```

```

UPDATE building SET infiltration_rate = 0.55, roof_u_value = 0.33
WHERE an_trans < 1900; --6542
UPDATE building SET infiltration_rate = 0.55, roof_u_value = 0.33
WHERE an_trans BETWEEN 1900 AND 1914; --6542
UPDATE building SET infiltration_rate = 0.55, roof_u_value = 0.33
WHERE an_trans BETWEEN 1915 AND 1945; --6542
UPDATE building SET infiltration_rate = 0.50, roof_u_value = 0.32
WHERE an_trans BETWEEN 1946 AND 1960; --2131
UPDATE building SET infiltration_rate = 0.40, roof_u_value = 0.32
WHERE an_trans BETWEEN 1961 AND 1970; --1104
UPDATE building SET infiltration_rate = 0.35, roof_u_value = 0.34
WHERE an_trans BETWEEN 1971 AND 1980; --759
UPDATE building SET infiltration_rate = 0.35, roof_u_value = 0.32
WHERE an_trans BETWEEN 1981 AND 1990; --893
UPDATE building SET infiltration_rate = 0.30, roof_u_value = 0.27
WHERE an_trans BETWEEN 1991 AND 2000; --1061
UPDATE building SET infiltration_rate = 0.30, roof_u_value = 0.20
WHERE an_trans BETWEEN 2001 AND 2010; --539
UPDATE building SET infiltration_rate = 0.30, roof_u_value = 0.17
WHERE an_trans > 2010; --200

```

```

select distinct(infiltration_rate) from building

```

```

UPDATE building SET frame_to_window_ratio = 0.25;

```

```

UPDATE building SET typehousing = NULL, occupancytype = NULL;
UPDATE building SET typehousing = 'Residential', occupancytype = 1
WHERE purpose like any(array['Hab plusieurs logements', 'Hab. -
rez activites', 'Habitation - activites', 'Habitation un
logement', 'Residence meublee', 'Hab. deux logements',
'Veranda']); --8755
UPDATE building SET typehousing = 'Office', occupancytype = 2
WHERE purpose like any(array['Bureaux des OIG', 'Bureaux',
'Administrations publiques']); --1053
UPDATE building SET typehousing = 'Garage', occupancytype = 3
WHERE purpose like any(array['Garage prive', 'Garage']); --1077
UPDATE building SET typehousing = 'Commercial', occupancytype = 4
WHERE purpose like any(array['Commerce', 'Depot', 'Atelier',
'Centre commercial']); --902

```

```

UPDATE building SET typehousing = 'Restaurant', occupancytype = 5
WHERE purpose like 'Restaurant'; --102
UPDATE building SET typehousing = 'Hotel', occupancytype = 6 WHERE
purpose like 'Hotel'; --155
UPDATE building SET typehousing = 'Hospital', occupancytype = 7
WHERE purpose like 'Hopital, Clinique'; --66
UPDATE building SET typehousing = 'Education', occupancytype = 8
WHERE purpose like any(array['Universite', 'College', 'Autre
ecole', 'Ecole privee', 'Jardin %', 'Ecole primaire', 'Salle de
sport']); --404
UPDATE building SET typehousing = 'Industry', occupancytype = 9
WHERE purpose like 'Usine'; --49
UPDATE building SET typehousing = 'Other', occupancytype = 10
WHERE typehousing is null; --2472

UPDATE building SET nopeople = 0 WHERE nopeople is null; --3686

UPDATE building SET window_u_value = 2.3, window_g_value = 0.47
WHERE gbauj < 2000;
UPDATE building SET window_u_value = 1.7, window_g_value = 0.49
WHERE gbauj >= 2000;

-- FOR REFURBISHMENT
UPDATE building SET window_u_value = 0.87, window_g_value = 0.6;
UPDATE building SET infiltration_rate = 0.4, roof_u_value = 0.17;

--FOR BIG REFURBISHMENT
UPDATE building SET composite_id_fk = 115;
UPDATE building SET infiltration_rate = 0.4, roof_u_value = 0.17;

select count(*) from building where zone = 9

select distinct(composite_id_fk) from building

UPDATE building SET blindslambda = 0.017, blindsirradiancecutoff =
400 WHERE occupancytype != 2;
UPDATE building SET blindslambda = 0.017, blindsirradiancecutoff =
200 WHERE occupancytype = 2;

--SOME OTHER BUILDINGS ARE UPDATED VISUALLY WITH GOOGLE EARTH
UPDATE building SET gastw = 5, n_floors = 5 WHERE egid = 1013255;
UPDATE building SET gastw = 5, n_floors = 5 WHERE egid = 2036983;
UPDATE building SET gastw = 4, n_floors = 4 WHERE egid =
295094208;
UPDATE building SET height = 9.0 WHERE egid = 295101591;

```



## Annex V : Java code to implement the solar PV on the roofs of the buildings.

```
//WRITING SOLAR PV PANELS AND HEATERS
if(solarPV) {
String myQuery5 = "SELECT panel_id FROM solar_pv where egid = " +
rs.getInt("egid") + " and scenario_id_fk = "+scenario+"";
Statement st5 = c.createStatement();
ResultSet rs5 = st5.executeQuery(myQuery5);

while (rs5.next()){
String myQuery6 = "SELECT * FROM data.citysim get pv(" +
rs5.getInt("panel_id") + ","+scenario+"";
System.out.println(myQuery6);
Statement st6 = c.createStatement();
ResultSet rs6 = st6.executeQuery(myQuery6);
while (rs6.next()) { // (should be only one line)
PGgeometry geom_pv, geom_pv_footprint;
double alpha = 0; //-rs6.getDouble("orientation")*3.1416/180;
double beta = 0; //-rs6.getDouble("inclination")*3.1416/180;
double roofTop =
(altitude+((double) (floorIndex))* (height/((double) (nfloors))));
double[][] orientationZ = {{Math.cos(alpha), -
Math.sin(alpha), 0}, {Math.sin(alpha), Math.cos(alpha), 0}, {0, 0, 1}};
double[][] orientationZinv = {{Math.cos(-alpha), -Math.sin(-
alpha), 0}, {Math.sin(-alpha), Math.cos(-alpha), 0}, {0, 0, 1}};
double[][] inclinationY = {{1, 0, 0}, {0, Math.cos(beta), -
Math.sin(beta)}, {0, Math.sin(beta), Math.cos(beta)}};
double[][] rotation = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}};
double[] XYZ1 = {0, 0, 0}, XYZ2 = {0, 0, 0};
for (int i = 0; i < inclinationY[0].length; i++) {
for (int j = 0; j < orientationZ[0].length; j++) {
for (int k = 0; k < inclinationY[0].length; k++) {
rotation[i][j] += inclinationY[i][k]*orientationZ[k][j];
}
}
}

PGgeometry centroidGeom = (PGgeometry)rs6.getObject("centroid");
Point centroid = (Point)centroidGeom.getGeometry();
double[] middlePt = {centroid.getX(), centroid.getY(), roofTop};
// if the footprint and ml_footprint exist
if ((rs6.getObject("ml_footprint") !=
null)&&(rs6.getObject("footprint")!=null)) {
geom_pv = (PGgeometry)rs6.getObject("ml_footprint");
geom_pv_footprint = (PGgeometry)rs6.getObject("footprint");
//System.out.println("Object: " +
geom_ground_roof.getGeometry().toString());

// read footprint and ml_footprint (in geom_ground_roof)
if ((geom_pv.getGeometry().getType() == Geometry.MULTILINESTRING)
&& // should not be necessary...
```

```

    (geom_pv_footprint.getGeometry().getType() ==
    Geometry.MULTIPOLYGON)) {

MultiLineString mls_pv = (MultiLineString)geom_pv.getGeometry();
MultiPolygon mpl_pv_footprint =
    (MultiPolygon)geom_pv_footprint.getGeometry();

// check geom have the same number of subgeom (zones)
//System.out.println(mls_ground_roof.toString());
if (mls_pv.numLines() != mpl_pv_footprint.numPolygons() ) {
System.err.println("Error : footprint and ls_footprint do not have
the same number of subgeometries.");
System.exit(1);
}
// for each zone
for (int lsZoneIndexPv = 0; lsZoneIndexPv < mls_pv.numLines();
lsZoneIndexPv++) {
//System.out.println("Polygon (zone): " + lsZoneIndexPv);

LineString ls_zone_pv = mls_pv.getLine(lsZoneIndexPv);
//fw.write("<Roof id=\"" + (surfaceIndex++) + "\" type=\""87\"
ShortWaveReflectance=\""0.2\" GlazingRatio=\""0.0\"
GlazingGValue=\""0.7\" GlazingUValue=\""1.4\"
OpenableRatio=\""0.0\">\n");
fw.write("<Surface id=\"" + (generate_id++) + "\"
ShortWaveReflectance=\""0.1\">\n");
fw.write("<PV pvRatio=\""0.98465\" Pmp=\""345\" Ac=\""1.604\"
Tref=\""25\" Tcnoct=\""41.5\" muVoc=\""-0.0837\" Vmp=\""28.65\"/>\n");
//fw.write("<solarHeater surfaceRatio=\""1.0\" eta0=\""0.778\"
a1=\""3.59\" a2=\""0.0096\"/>\n");

double minZ = 99999999;
for( int p = 0; p < ls_zone_pv.numPoints()-1; p++ ) {
Point pt = ls_zone_pv.getPoint(p);
XYZ1[0] = pt.getX()-middlePt[0];
XYZ1[1] = pt.getY()-middlePt[1];
XYZ1[2] = 0;

XYZ2[0] =
XYZ1[0]*rotation[0][0]+XYZ1[1]*rotation[0][1]+XYZ1[2]*rotation[0][
2];
XYZ2[1] =
XYZ1[0]*rotation[1][0]+XYZ1[1]*rotation[1][1]+XYZ1[2]*rotation[1][
2];
XYZ2[2] =
XYZ1[0]*rotation[2][0]+XYZ1[1]*rotation[2][1]+XYZ1[2]*rotation[2][
2];

XYZ1[0] =
XYZ2[0]*orientationZinv[0][0]+XYZ2[1]*orientationZinv[0][1]+XYZ2[2
]*orientationZinv[0][2];
XYZ1[1] =
XYZ2[0]*orientationZinv[1][0]+XYZ2[1]*orientationZinv[1][1]+XYZ2[2
]*orientationZinv[1][2];

```

```

XYZ1[2] =
XYZ2[0]*orientationZinv[2][0]+XYZ2[1]*orientationZinv[2][1]+XYZ2[2]
]*orientationZinv[2][2];

if(minZ >= XYZ1[2]){
minZ = XYZ1[2];
}
}

for( int p = 0; p < ls_zone_pv.numPoints()-1; p++ ) {
Point pt = ls_zone_pv.getPoint(p);
XYZ1[0] = pt.getX()-middlePt[0];
XYZ1[1] = pt.getY()-middlePt[1];
XYZ1[2] = 0;

XYZ2[0] =
XYZ1[0]*rotation[0][0]+XYZ1[1]*rotation[0][1]+XYZ1[2]*rotation[0][
2];
XYZ2[1] =
XYZ1[0]*rotation[1][0]+XYZ1[1]*rotation[1][1]+XYZ1[2]*rotation[1][
2];
XYZ2[2] =
XYZ1[0]*rotation[2][0]+XYZ1[1]*rotation[2][1]+XYZ1[2]*rotation[2][
2];

XYZ1[0] =
XYZ2[0]*orientationZinv[0][0]+XYZ2[1]*orientationZinv[0][1]+XYZ2[2]
]*orientationZinv[0][2]+middlePt[0];
XYZ1[1] =
XYZ2[0]*orientationZinv[1][0]+XYZ2[1]*orientationZinv[1][1]+XYZ2[2]
]*orientationZinv[1][2]+middlePt[1];
XYZ1[2] =
XYZ2[0]*orientationZinv[2][0]+XYZ2[1]*orientationZinv[2][1]+XYZ2[2]
]*orientationZinv[2][2]+roofTop-minZ+1;

fw.write("<V" + p + " x=\"" + XYZ1[0] + "\" y=\"" + XYZ1[1] + "\"
z=\"" + XYZ1[2] + "\"/>\n");
}

fw.write("</Surface>\n");

//fw.write("</Roof>\n");
}
}
}
//System.out.println("Object: " + rotation[2][1] + +
rotation[2][2]);
}
}
}

```







## Annex VI : MATLAB code to evaluate the thermal comfort as a function of the heat tank volume and the HP electrical rated power.

```
javaclasspath('postgresql-9.4.1207.jar','postgis-jdbc-2.2.0dev.jar');

command = sprintf('mingw32-make -f MakefileXML.mingw32 MAINCLASS=CitySimDatabaseLinkerXML run');
[status,cmdout] = system(command)
command = 'CitySimPro essai.xml';
[status,cmdout] = system(command)
command = sprintf('mingw32-make -f MakefileReader.mingw32 MAINCLASS=CitySimDatabaseLinkerReader2 run');
[status,cmdout] = system(command)

props=java.util.Properties;
props.setProperty('user', 'kaempf');
props.setProperty('password', 'jerome');

% Create the database connection (port 5432 is the default
% postgres chooses
% on installation)
driver=org.postgresql.Driver;
url = 'jdbc:postgresql://128.178.6.45:5432/geneva';
conn=driver.connect(url, props);

query='select egid, qs2 from models where scenario_id_fk = 0 order
by egid desc'; % Gets all records
st=conn.createStatement();
rs=st.executeQuery(query);
k = 1;
while (rs.next())
    results2 = strsplit(char(rs.getString('qs2')),{' ',' ',' ',' '});
    q2(k, :) = str2double(results2(2:8761));
    results0 = char(rs.getString('egid'));
    egid(k,1) = str2double(results0);
    k = k + 1;
end

for i=1:1:365
    q2_day(:,i) = sum(q2(:, (i-1)*24+1:i*24))';
end

qmax_hour = max(q2)';
qmax_day = max(q2_day)';
volumes = 3600*qmax_day/(1000*1000*4.18*(35-21))

n=2;
% vol = [volumes*0.04:volumes*0.02:0.6*volumes];
```

```

% power = [qmax_hour*0.1:qmax_hour*0.01:qmax_hour*0.45];
vol = [volumes*0.025:volumes*0.1187:0.5*volumes];
power = [qmax_hour*0.2:qmax_hour*0.0025:qmax_hour*0.46];
PMV = zeros(length(vol),length(power));
PPD = zeros(length(vol),length(power));
Tair = zeros(length(vol),length(power));
for i=[1:1:length(vol)]
    for j=[1:1:length(power)]
        loss
= (4*(2*3.1416*(vol(i)/(2*3.1416))^(2/3)+(4*3.1416*(vol(i)/(2*3.141
6))^(2/3)))*0.29307/0.3048^2)/(33-21); %5 walls * surface * 5
BTU/(ft2*h)*0.293 W/BTUh / 0.304^2m^2/ft^2

        A =
textread('CitysimDatabaseLinkerXML.java','%s','delimiter','\n',
'bufsize',2147483647);

        A{7} = sprintf('public class
CitySimDatabaseLinkerXML%d',n);
        A{214} = sprintf('double vol = %d;',vol(i));
        A{215} = sprintf('double loss = %d;',loss);
        A{216} = sprintf('double power = %d;',power(j));

        name = sprintf('CitysimDatabaseLinkerXML%d.java',n);
        [fileID, message] = fopen(name,'wt+');
        for k = 1:1:length(A)
            fprintf(fileID,'%s\n',A{k});
        end
        fclose(fileID);

        command = sprintf('mingw32-make -f MakefileXML.mingw32
MAINCLASS=CitySimDatabaseLinkerXML%d run',n);
        [status,cmdout] = system(command)
        command = 'CitySimPro essai.xml';
        [status,cmdout] = system(command)
        command = sprintf('mingw32-make -f MakefileReader.mingw32
MAINCLASS=CitySimDatabaseLinkerReader run');
        [status,cmdout] = system(command);

        % Create the database connection (port 5432 is the default
postgres chooses
        % on installation)
        query='select ta, pe, pm, qh from comfort where
scenario_id_fk = 0 order by egid desc'; % Gets all records
        st=conn.createStatement();
        rs=st.executeQuery(query);
        k = 1;
        while (rs.next())
            results =
strsplit(char(rs.getString('ta')),{'{','}',',',' '});
            Ta1(k,:) = str2double(results(2:8761));
            Qh(i,j) = rs.getDouble('qh');
            Pe(i,j) = rs.getDouble('pe'); %In kWh

```

```

        Pm(i,j) = rs.getDouble('pm')/1000; %In kWh
        k = k + 1;
    end
    [buildings, hours] = size(Ta1);
    %Source file:///C:/Users/TEMP.INTRANET.001/Downloads/2.pdf
    %Initializing the parameters of Fanger's equation
    for u=[1:1:buildings]
        for v=[1:1:hours]
            alpha=0.25;
            M = 100; %Metabolic rate in W/m2
            V = 0; %Effective mechanic power
            Icl = 0.5*0.155; %Intrinsic clothing insulation in
K*m^2/W
            RH = 50; %Humidity as a %
            Ta = Ta1(u,v); %Air temperature
            Pa = (10*RH)*exp(16.6536-4030.183/(Ta + 235)); %In
kPa
            Tg = Ta; %Globe termometer temperature in °C
            Ts = Ta; %Dry thermometer temperature
            Va = 0.1; %Air speed in m/s
            Trm = Tg+1.9*sqrt(Va)*(Tg-Ts);
            hcl = 12.1*sqrt(Va);
            if Icl<=0.078
                fcl = 1+1.290*Icl;
            else
                fcl = 1.05+0.645*Icl;
            end
            tcl = 10;
            tcl2=tcl;
            hcl2=hcl;
            for w=[1:120]
                tcl = 35.7-0.0275*(M-V)-Icl*(3.96*10^(-
8)*fcl*((tcl+273)^4-(Trm+273)^4)+fcl*hcl*(tcl-Ta));
                if 2.05*abs(tcl-Ta)^0.25 > 12.1*sqrt(Va)
                    hcl = 2.38*abs(tcl-Ta)^0.25; %Convection
coefficient in W/(m^2*k)
                else
                    hcl = 12.1*sqrt(Va);
                end
                tcl=(1-alpha)*tcl+alpha*tcl2;
                hcl=(1-alpha)*hcl+alpha*hcl2;
                tcl2=tcl;
                hcl2=hcl;
            end
            PMV2 = -0.04+(0.303*exp(-0.036*M)+0.028)*((M-V)-
3.05*10^(-3)*(5733-6.99*(M-V)-Pa)-0.4*((M-V)-58.15)-1.7*10^(-
5)*M*(5867-Pa)-0.0014*M*(34-Ta)-3.96*10^(-8)*fcl*((tcl+273)^4-
(Trm+273)^4)-fcl*hcl*(tcl-Ta));
            PPD(i,j) = PPD(i,j)+(100-95*exp(-0.03353*PMV2^4-
0.2179*PMV2^2))/8760;
            PMV(i,j)=PMV(i,j)+PMV2/8760;
            Tair(i,j)=Tair(i,j)+Ta/8760;
        end
    end
end

```

```

        i
    end
end

CostPower = [5000 7000 10000; 0.904*(3250+3590)/2
0.904*(4250+4550)/2 0.904*(5110+5630+6890)/3];
CostHeatTank = [3 3.3 5 7.4 10 15.1 15.5 18 22.46 30;1550 1522
1734 2370 2521 2825 2877 3846 3950 5660]; % *1.1 for installation
cost and /1.1 for euro dollar conv

yearsSpace = 999;
yearsTank = 30;
yearsHP = 20;
i = 0.06;

IspaceTank = 3.1416*(vol/(2*3.1416)).^(2/3)*8000/1.086;
IspaceTankAnnualized =
IspaceTank*(i*(1+i)^yearsSpace)/((1+i)^yearsSpace-1);

Itank =
interp1(CostHeatTank(1,:),CostHeatTank(2,:),vol,'linear','extrap')
ItankAnnualized = Itank*(i*(1+i)^yearsTank)/((1+i)^yearsTank-1);

Ihp =
interp1(CostPower(1,:),CostPower(2,:),power,'cubic','extrap')
IhpAnnualized = Ihp*(i*(1+i)^yearsHP)/((1+i)^yearsHP-1);

for k=[1:1:length(vol)]
    for j=[1:1:length(power)]
        CAPEX(k,j) = IspaceTankAnnualized(k) + ItankAnnualized(k)
+ IhpAnnualized(j); %*(8000/1.0859) polyethylene insulation maybe
        OPEX(k,j) = Pe(k,j)*0.21/1.086; %0.21 fr/kWh
        Total(k,j) = CAPEX(k,j)+OPEX(k,j);
    end
end

[X,Y] = meshgrid(power/1000, vol);
surf(X,Y,(100-PPD))
title('Thermal comfort')
ylabel('Heat tank volume [m^3]')
xlabel('Electrical power of HP [kW]')
zlabel('% of people satisfied')

```



## Annex VII: MATLAB code to compare the different types of models

```
javaclasspath('postgresql-9.4.1207.jar','postgis-jdbc-2.2.0dev.jar');

props=java.util.Properties;
props.setProperty('user','kaempf');
props.setProperty('password','jerome');

% Create the database connection (port 5432 is the default
% postgres chooses
% on installation)
driver=org.postgresql.Driver;
url = 'jdbc:postgresql://128.178.6.45:5432/geneva';
conn=driver.connect(url, props);

query='select egid, ta1, ta2, ta3, ta4, qi1, qi2, qi3, qi4, qs1,
qs2, qs3, qs4 from models where scenario_id_fk = 0 order by egid
desc'; % Gets all records
st=conn.createStatement();
rs=st.executeQuery(query);

% egidNew = [295168134 1013232 2039264 295102607 2039711 1013231
1013233 1013242 2039265, 2037241 2038216 2038908 2037118 2038577
2038553 2038123 1010433 2039502 1010142 2039612 2038442 2038583
295097020 295105482 2038852 1010137 2037239 1010429 295009964
2038905 1010179 2038796 2038552 2038550 295098523];

i = 1;
while (rs.next())
    results0 = char(rs.getString('egid'));
    egid(i,1) = str2double(results0);

    results1 = strsplit(char(rs.getString('ta1')),{' ',' ',' ',' '});
    ta1(i,:) = str2double(results1(2:8761));

    results2 = strsplit(char(rs.getString('ta2')),{' ',' ',' ',' '});
    ta2(i,:) = str2double(results2(2:8761));

    results3 = strsplit(char(rs.getString('ta3')),{' ',' ',' ',' '});
    ta3(i,:) = str2double(results3(2:8761));

    results10 = strsplit(char(rs.getString('ta4')),{' ',' ',' ',' '});
    ta4(i,:) = str2double(results10(2:8761));

    results4 = strsplit(char(rs.getString('qi1')),{' ',' ',' ',' '});
    qi1(i,:) = str2double(results4(2:8761));

    results5 = strsplit(char(rs.getString('qi2')),{' ',' ',' ',' '});
    qi2(i,:) = str2double(results5(2:8761));
```

```

        results6 = strsplit(char(rs.getString('qi3')),{'{','}',' ',' '});
        Qi3(i,:) = str2double(results6(2:8761));

        results10 = strsplit(char(rs.getString('qi4')),{'{','}',' ',' '});
        Qi4(i,:) = str2double(results10(2:8761));

        results7 = strsplit(char(rs.getString('qs1')),{'{','}',' ',' '});
        Qs1(i,:) = str2double(results7(2:8761));

        results8 = strsplit(char(rs.getString('qs2')),{'{','}',' ',' '});
        Qs2(i,:) = str2double(results8(2:8761));

        results9 = strsplit(char(rs.getString('qs3')),{'{','}',' ',' '});
        Qs3(i,:) = str2double(results9(2:8761));

        results10 = strsplit(char(rs.getString('qs4')),{'{','}',' ',' '});
        Qs4(i,:) = str2double(results10(2:8761));

        i = i + 1
end

[buildings,periods] = size(Qs1);

egidOrdered =
[1013176,1013177,1013173,1013174,1013175,2037903,295081234,2951035
33,295101571,1013295,1012904,1013294,295075801,1012906,2037077,101
2907,1013284,1013283,1013293,1013291];

Figure(1)
subplot(2,1,1)
k=1;
for j=1:2.5:2.5*(buildings)
    i = find(egid==egidOrdered(k));
    bar(j,100*(sum(Qs2(i,:))-sum(Qs1(i,:)))/sum(Qs1(i,:)),'r')
    hold on
    bar(j+1,100*(sum(Qs3(i,:))-sum(Qs1(i,:)))/sum(Qs1(i,:)),'g')
    hold on
    k=k+1;
end
title(sprintf('Annual heat demand difference with respect to Model
1\n(Qh_i-Qh_1)/Qh_1'));
xlabel('Buildings')
ylabel('Annual heat demand difference [%]')
legend('Model 2','Model 3');
% set(ax,'YLim',[0.0 range])
ylim([0 10])
xlim([0 j+3]);
% set(ax2,'XTick',[0.0:0.01:j+2])
hold off

subplot(2,1,2)
k=1

```

```

for j=1:2.5:2.5*(buildings)
    i = find(egid==egidOrdered(k));
    bar(j,100*(sum(Qi2(i,:))-sum(Qi1(i,:)))/sum(Qi1(i,:)),'r')
    hold on
    bar(j+1,100*(sum(Qi3(i,:))-sum(Qi1(i,:)))/sum(Qi1(i,:)),'g')
    hold on
    k=k+1;
end
title(sprintf('Annual internal gains difference with respect to
Model 1\n(Qi_i-Qi_1)/Qi_1'));
xlabel('Buildings')
ylabel('Annual internal gains difference [%]')
legend('Model 2','Model 3');
% set(ax,'YLim',[0.0 range])
ylim([0 2])
xlim([0 j+3]);
% set(ax2,'XTick',[0.0:0.01:j+2])
hold off

```

Figure (2)

```

m=0;
for l=[1 9] %1:1:2
%     Figure(l+1)
    ta1_day=zeros(1,24);
    ta2_day=zeros(1,24);
    ta3_day=zeros(1,24);
    k=find(egid==egidOrdered(l));
    n=0;
    m=1+m;
    subplot(3,2,m)
    for i=1:1:24
        for j=[1:1:365]
            ta1_day(i)=ta1_day(i)+ta1(k,i+24*(j-1))/365;
            ta2_day(i)=ta2_day(i)+ta2(k,i+24*(j-1))/365;
            ta3_day(i)=ta3_day(i)+ta3(k,i+24*(j-1))/365;
        end
        bar(n,ta1_day(i))
        hold on
        bar(n+1,ta2_day(i),'r')
        hold on
        bar(n+2,ta3_day(i),'g')
        hold on
        n=4+n;
    end
    hold off
    title(sprintf('Hourly temperature'));
    xlim = [1 24];
    xlabel('hours')
    ylabel('Hourly average Temperature °C')
    legend('Model 1','Model 2','Model 3');
    Qs1_day=zeros(1,24);
    Qs2_day=zeros(1,24);
    Qs3_day=zeros(1,24);
    n=0;

```

```

subplot(3,2,m+2)
for i=1:1:24
    for j=[1:1:365]
        Qs1_day(i)=Qs1_day(i)+Qs1(k,i+24*(j-1))/365;
        Qs2_day(i)=Qs2_day(i)+Qs2(k,i+24*(j-1))/365;
        Qs3_day(i)=Qs3_day(i)+Qs3(k,i+24*(j-1))/365;
    end
    bar(n,Qs1_day(i))
    hold on
    bar(n+1,Qs2_day(i),'r')
    hold on
    bar(n+2,Qs3_day(i),'g')
    hold on
    n=4+n;
end
hold off
title(sprintf('Daily average heat supplied (Wh)'));
xlim = [1 24];
xlabel('hours')
ylabel('Daily average heat supplied (Wh)')
legend('Model 1','Model 2','Model 3');

Qi1_day=zeros(1,24);
Qi2_day=zeros(1,24);
Qi3_day=zeros(1,24);
n=0;
subplot(3,2,m+4)
for i=1:1:24
    for j=[1:1:365]
        Qi1_day(i)=Qi1_day(i)+Qi1(k,i+24*(j-1))/365;
        Qi2_day(i)=Qi2_day(i)+Qi2(k,i+24*(j-1))/365;
        Qi3_day(i)=Qi3_day(i)+Qi3(k,i+24*(j-1))/365;
    end
    bar(n,Qi1_day(i))
    hold on
    bar(n+1,Qi2_day(i),'r')
    hold on
    bar(n+2,Qi3_day(i),'g')
    hold on
    n=4+n;
end
hold off
title(sprintf('Daily average internal gains (Wh)'));
xlim = [1 24];
xlabel('hours')
ylabel('Daily average internal gains (Wh)')
legend('Model 1','Model 2','Model 3');
egid(k)
end

```



