The Journal of Open Source Software

CESAR-P: A dynamic urban building energy simulation tool

Kristina Orehounig [1¶], **Leonie Fierz** [*1], **James Allan** [1], **Sven Eggimann** [1], **Natasa Vulic** [1], and **Aaron Bojarski** [1]

1 Urban Energy Systems Laboratory, Empa, Duebendorf, Switzerland ¶ Corresponding author

DOI: 10.21105/joss.04261

Software

- Review
- Repository
- Archive

Editor: Tim Tröndle

Reviewers:

- @olejandro
- @noah80
- @jasondegraw

Submitted: 06 March 2022
Published: 12 October 2022

Summary

Buildings are responsible for a large share of energy consumption and carbon dioxide emissions. With the challenge of reducing energy consumption and integrating renewable energy sources in buildings and neighbourhoods, an understanding of energy consumption patterns is needed at different temporal and spatial scales. CESAR-P (Combined Energy Simulation And Retrofitting - Python) is a bottom-up building stock modelling software to evaluate energy consumption and retrofitting strategies for individual buildings and neighbourhoods. CESAR-P is used to parameterize models for the dynamic building energy simulation tool EnergyPlus (Crawley et al. (2001)). It computes hourly energy demand profiles and indoor temperatures at the building level and takes into account shading and reflections of surrounding buildings. After the current energy demand is computed, retrofitting measures for individual buildings can be evaluated in terms of energy savings, embodied emissions and costs. The default library of CESAR-P is based on Swiss archetypal buildings created from standards and statistics about the Swiss building stock; however, users are able to define their own constructions and internal conditions for their buildings. CESAR-P is written in Python and is a further development of the tool CESAR (Wang et al. (2018)). The building model generator of CESAR-P creates EnergyPlus input data files (.IDF) for each building within a specified extent, and can process geo-referenced input data as shapefiles (ESRI .shp format) for geometry. Building usage type and construction information is used to populate the building model. The EnergyPlus files are executed with a weather file (EnergyPlus epw weather data format) of the geographic location which provides the necessary climatic context for the building. The results consist of a configurable set of yearly demand values and hourly resolved times series for each building from EnergyPlus and additional operational costs and emissions. The retrofit module can be used to update the building models according to a specified retrofit strategy. Retrofit strategies specify the frequency of retrofits to the main building elements such as walls, windows and roofs across the building stock. The output of the retrofit module include, in addition to the above-mentioned operational indicators, costs and embodied emissions of retrofitting interventions. All simulation steps can also be run in parallel on multiple cores. The code is platform-independent and is tested to run on Linux and Windows. The source code for CESAR-P has been archived to Zenodo: (Fierz & UrbanEnergySystemsLab, 2021)

Statement of need and Key features

Building energy simulation is used to generate the expected energy demand profiles of buildings. Building energy demand profiles are useful for planners to evaluate current performance and investigate how buildings will perform under different scenarios (e.g. building retrofits, changing climate). Measured demands are often unavailable which makes building energy

*co-first author

simulation a key tool for decision-making. Urban building energy modelling (UBEM) has been established in recent years to support the implementation of energy conservation measures and the integration of renewable energy technologies at city and district scale. UBEM tools can be classified as taking a top-down or a bottom-up modelling approach. Top-down approaches rely on aggregated data and different drivers (e.g. socio-economic indicators or climate data), whereas bottom-up modelling approaches are based on data at an individual building scale and include either statistical or engineering based modelling techniques to compute energy demand data of a set of individual buildings (Johari et al., 2020). Various review papers summarize available tools, including their key-features and limitations (Johari et al., 2020). The here presented tool CESAR-P falls within the category of bottom-up modelling approaches. CESAR-P uses EnergyPlus as the simulation engine, an open-source building energy simulation software that is widely-used in both academia and industry. CESAR-P simplifies the creation of EnergyPlus input files using a set of key parameters which are stored in archetypes. The target audience of the software are energy planners, energy utilities, cities and researchers. The time-resolved energy consumption of buildings at large scale that CESAR-P simulates are crucial for developing renewable energy integration and retrofit strategies at different scales, from the building to the neighborhood and city scale. Key features of CESAR-P include:

- Modelling of indoor temperatures, heating and cooling loads, domestic hot water consumption, electricity consumption, comfort parameters, operational costs and emissions, embodied emissions and investment costs of retrofitting solutions, and solar irradiance on external surfaces.
- Capability to automate extraction of building geometries from commonly used geometric file types (such as shapefiles), simplifying the generation of required input files.
- The archetypes used to parameterize the building models are stored as RDF triples (in TTL files). The triples contain ABox statements related to the construction properties defined by each archetype. The statements conform to classes and properties stored in the urban energy simulation ontology (Allan et al. (2021)). By default, a construction archetype will be assigned to a building instance using the building age. The construction archetype contains all of the material layers in the floor, roof and external walls of the building. CESAR-P loads the archetype data directly from the provided TTL files, but it can be configured to access a graph database instance to query archetype data. The reason to store the source data using RDF is due to the growing number of UBEM studies. It has been found that it is challenging to track assumptions used in published studies. The use of semantic web technologies such as RDF enables the user to query the results and underlying data used to build the simulation inputs. The use of ontologies will enable the linking of concepts and data models used in other studies using TBox inferences.
- Evaluation of retrofitting options and strategies.
- Simulation and comparison of energy demand for scenarios with different building parametrization, for example to evaluate passive cooling potential (Silva et al. (2022)).
- Since CESAR-P uses EnergyPlus, which is a continually developed energy simulation tool within the research field, the capabilities of CESAR-P can be extended to accommodate future developments and improve modelling accuracy.
- The tool can be applied at various scales and can be used for predicting the energy consumption of a single building, of whole neighbourhoods with thousands of buildings (Wang et al. (2018)) or, in combination with clustering approaches, for calculating national building energy demands (Murray et al. (2020), Eggimann et al. (2022)).

Example

Following we are outlining how you set up a simulation with cesar-p-core library. The snippets are based on the basic usage example in cesar-p-usage-examples.

The main API class of `cesar-p-core` library is the `SimulationManager`. On initialization of the class you need to pass the configuration and the output folder for your project. Additionally, a process wide instance of the unit registry (handling conversion of physical units) must be provided. A minimal code-snippet looks like:

```python
import cesarp.common
from cesarp.manager.SimulationManager import SimulationManager

if __name__ == "__main__":
    sim_manager = SimulationManager(base_output_path="results",
                                    main_config="simple_main_config.yml",
                                    unit_reg=cesarp.common.init_unit_registry())
    sim_manager.run_all_steps()
```

In the YAML configuration you need to define the input files of your project. Below is the minimal set of YAML configuration you need for any project. The code above refers to this file as `simple_main_config.yml`.

```yaml
MANAGER:
    SITE_VERTICES_FILE:
        PATH: "SiteVertices.csv"
    BLDG_FID_FILE:
        PATH: "BuildingInformation.csv"
        SEPARATOR: ","
        LABELS:
            gis_fid: "ORIG_FID"
    BLDG_AGE_FILE:
        PATH: "BuildingInformation.csv"
    BLDG_TYPE_PER_BLDG_FILE:
        PATH: "BuildingInformation.csv"
    SINGLE_SITE:
        WEATHER_FILE: "Zurich_2015.epw"
```

The `SiteVertices.csv` defines the footprints and height of all the buildings on your site.

TARGET_FID	POINT_X	POINT_Y	HEIGHT
1	0	0	12.5
1	0	0	12.5
1	25	0	12.5
1	25	0	12.5
1	0	0	12.5
2	35	0	12.5
2	35	0	12.5
2	60	0	12.5
2	60	0	12.5
2	35	0	12.5
…			

The `BuildingInformation.csv` file defines the properties of the buildings. The configuration entry `BLDG_FID_FILE` requires the column `ORIG_FID`, entry `BLDG_TYPE_PER_BLDG_FILE` column `SIA2024BuildingType` and entry `BLDG_AGE_FILE` column `BuildingAge`. This example above includes a few more columns per building, which will be explained further down. Usually you will have one file defining all the properties of your buildings, but the configuration gives you the flexibility to point to separate files for the different building properties and also to map to custom column names to allow using input data files without pre-processing (see the sub-entries for `BLDG_FID_FILE`).

ORIG_FID	SIA2024BuildingType	BuildingAge	ECarrierHeating	ECarrierDHW	GlazingRatio
1	MFH	1900	2	2	13
2	MFH	1930	2	2	16
…					
8	MFH	2012	2	2	30
9	MFH	2015	2	2	30

The last required input is the weather data in EnergyPlus epw format, in the example named `Zurich_2015.epw`.

Building model parametrization

The main source for constructional parameters are the archetype definitions (included in the library) and assigned by the age of a building. Operational parameters such as occupancy or loads from appliances are based on a Swiss norm (SIA2024). A few further parameters and constants can be set in the YMAL configuration. You find an overview of all the parameters you can specify in the YAML in cesar-p-config-overview.yaml. You only need to overwrite the properties you want to change in your project specific YAML configuration. The configuration entries should be placed under the correct category, while paying attention to the indentation. The example below overwrites the following properties:

- `RANDOM_CONSTRUCTIONS` if set to `True`, CESAR-P chooses randomly one of the available construction archetypes, if there are several matching the age class of the processed building.
- `GLAZING_RATIO_PER_BLDG_FILE` by default the glazing ratio is defined by a lookup based on the age of the buildindg. If you have information of the glazing ratio per building, add this information e.g. to your `BuildingInformation.csv` under a column heading `GlazingRatio`.
- `RADIUS` defines the radius around a building in which other buildings are simulated as shading objects
- `MINIMAL_STORY_HEIGHT` defines the default minimum floor height to use. The number of floors is defined by the integer division of building height and MINIMAL_STORY_HEIGHT.

Your YAML configuration including those parameters would look like:

```
MANAGER:
    # here go first all the default parameters from above
    ....
    RANDOM_CONSTRUCTIONS: True
    GLAZING_RATIO_PER_BLDG_FILE:
        ACTIVE: True
        PATH: "BuildingInformation.csv"
GEOMETRY:
    NEIGHBOURHOOD:
        RADIUS: 50 # meter
MAIN_BLDG_SHAPE:
    MINIMAL_STORY_HEIGHT: 2.2 # meter
```

If you need even more customization, you can plug in your own implementation for assignment of the constructional and operational parameters per building. See custom_constr_archetype_mapping and operation_params_per_floor, respectively, in cesar-p-usage-examples repository.

Results

The results consist of a set of parameters from EnergyPlus results in a table format, as absolute annual values and converted to specific per m2 values. Additionally you can choose to calculate operational emissions. For this, you need to specify the energy installation per building. In the configuration, point to the file containing the data and activate the feature:

```
MANAGER:
    # here go first all the default parameters from above
    ...
    DO_CALC_OP_EMISSIONS_AND_COSTS: True
    BLDG_INSTALLATION_FILE:
        PATH: "../example_project_files/BuildingInformation.csv"
```

The reuqired columns in the linked file are `ECarrierHeating`, `ECarrierDHW` defining the number key of the energy source used for space heating and domestic hot water, respectively. The allowed options are defined in the enumeration EnergySource.

The table below is an extract of the `site_result_summary.csvy` showing most important annual results per building:

Bldg	Heating Annual specific	DHW Annual specific	Electricity Annual specific	Cooling Annual specific	Total PEN	Total CO2
	kWh/m2 /year	kWh/m2 /year	kWh/m2 /year	kWh/m2 /year	kWh/m2 /year	Oileq * MJ/m2 /year
1	101	18	23	0.9	898	49
2	113	18	23	1.1	963	54
…						

Full list of result values reported is shown in the table below:

Variable	Unit
EnergyPlus error level	-
Heating Annual	kWh/year
DHW Annual	kWh/year
Electricity Annual	kWh/year
Cooling Annual	kWh/year
Heating Annual specific	kWh/year/m2
DHW Annual specific	kWh/year/m2
Electricity Annual specific	kWh/year/m2
Cooling Annual specific	kWh/year/m2
Total bldg floor area	m2
Total PEN	Oileq * MJ/m2/year
PEN Heating	Oileq * MJ/m2/year
PEN DHW	Oileq * MJ/m2/year
PEN Elec	Oileq * MJ/m2/year
Total CO2	CO2eq * kg/m2/year
CO2 Heating	CO2eq * kg/m2/year
CO2 DHW	CO2eq * kg/m2/year
CO2 Elec	CO2eq * kg/m2/year
Heating - Fuel Costs	CHF/year
DHW - Fuel Costs	CHF/year
Electricity - Costs	CHF/year

Variable	Unit
Year used for lookup of cost factors	-

It is a good practice to check the column `EnergyPlus error level` to make sure there are no errors or warnings from EnergyPlus simulation. If there are, the details of the error are provided in the `eplus_error_summary.err` file.

In case the details about the building model geometry are desired, they are available in the `bldgs_infos_model_generation.csv` file. For example, one can find the number of stories and the resulting floor height of each building's geometry, as well as information on the effective glazing ratio, which can differ from your setting in case there are many small walls not having space for a window.

If hourly results are needed, adding the following line to the main script would output the domestic hot water and heating hourly values as a pandas data frame:

Additional imports needed

```
from cesarp.eplus_adapter.eplus_eso_results_handling
    import RES_KEY_DHW_DEMAND, RES_KEY_HEATING_DEMAND
from cesarp.eplus_adapter.idf_strings import ResultsFrequency
```

After the simulation run, you can qurey the results form your `SimulationManager` instance:

```
hourly_demand =
    sim_manager.collect_custom_results(
      result_keys=[RES_KEY_HEATING_DEMAND, RES_KEY_DHW_DEMAND],
      results_frequency=ResultsFrequency.HOURLY)
```

Retrofit

For retrofitting the general procedure is to first define a baseline simulation and then run on top one or more retrofit scenarios. There are two retrofit managers implemented, the basic SimpleRetrofitManager and EnergyPerspective2050RetrofitManager, following more complex rules to decide on retrofit measures. These manager classes are used instead of the `SimulationManager` to run your simulations. One can also implement a custom retrofit manager that defines which retrofit measures are carried out on which buildings.

A summary of the simulation results of each scenario is saved to `all_scenarios_result_summary.csvy`. As an example the table below shows a comparison of annual specific heating demand for three different scenarios, `orig` being the baseline, scenario `roof` where roof construction was retrofitted and `wall_win_groundf` where wall, windows and ground floor were retrofitted. Details about retrofit measures, costs and embodied emissions is available in `retrofit_log.csv` in each scenario result subfolder. Summarized figures out of those files are included in the table.

Building	Scenario	Heating Annual specific	Retrofit Costs	Retrofit embodied CO2 emissions
		kWh/m2/year	CHF	CO2eq * kg
1	orig	101	-	-
1	roof	94	150'000	3'389
1	wall_win_groundf	18	334'077	21'831
2	orig	113	-	-
2	roof	106	150'000	10'067
2	wall_win_groundf	18	349'065	22'560

See the retrofit example for more details.

Project handling

There are several features to simplify project management, including comparison between different simulation runs with changed parameters, re-loading existing projects to avoid re-generation of building models or simulating existing IDF files. To allow for exchanging or backing up projects, there is the option to save all input files including the constructions database in a ZIP file. Check out the basic usage example to see those features in action.

Acknowledgements

The development of the software CESAR-P has been financially supported by SCCER FEEB&D (CTI.1155000149) and by Empa. We acknowledge contributions from Danhong Wang, Jonas Landolt, Georgios Mavromatidis and Portia Murray to CESAR, the predecessor version of CESAR-P. We would also like to acknowledge Aaron Bojarski who supported the development of CESAR-P and the development of the urban energy simulation ontology. Further we would like to acknowledge Ricardo Parreira da Silva for the implementation of passive cooling features.

References

Allan, J., Fierz, L., Bollinger, A., & Orehounig, K. (2021). Linked data ontology for urban and national scale building energy modelling. *eSim Conference Proceedings*. http://www.ibpsa.org/proceedings/eSimPapers/2021/Contribution_1148_final_a.pdf

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., Witte, M. J., & Glazer, J. (2001). EnergyPlus: Creating a new-generation building energy simulation program. *Energy and Buildings*, *33*(4), 319–331. https://doi.org/10.1016/S0378-7788(00)00114-6

Eggimann, S., Vulic, N., Rüdisüli, M., Mutschler, R., Orehounig, K., & Sulzer, M. (2022). Spatiotemporal upscaling errors of building stock clustering for energy demand simulation. *Energy and Buildings*, 111844. https://doi.org/10.1016/j.enbuild.2022.111844

Fierz, L., & UrbanEnergySystemsLab. (2021). *Hues-platform/cesar-p-core*. Zenodo. https://doi.org/10.5281/zenodo.4682880

Johari, F., Peronato, G., Sadeghian, P., Zhao, X., & Widen, J. (2020). Urban building energy modeling: State of the art and future prospects. *Renewable and Sustainable Energy Reviews*, *128*, 109902. https://doi.org/10.1016/j.rser.2020.109902

Murray, P., Marquant, J., Niffeler, M., Mavromatidis, G., & Orehounig, K. (2020). Optimal transformation strategies for buildings, neighbourhoods and districts to reach CO2 emission reduction targets. *Energy and Buildings*, *207*, 109569. https://doi.org/10.1016/j.enbuild.2019.109569

Silva, R., Eggimann, S., Fierz, L., Fiorentini, M., Orehounig, K., & Baldini, L. (2022). Opportunities for passive cooling to mitigate the impact of climate change in switzerland. *Buildings and Environment*, 108574. https://doi.org/10.1016/j.buildenv.2021.108574

Wang, D., Landolt, J., Mavromatidis, G., Orehounig, K., & Carmeliet, J. (2018). CESAR: A bottom-up building stock modelling tool for swit-zerland to address sustainable energy transformation strategies. *Energy and Buildings*, *169*, 9–26. https://doi.org/10.1016/j.enbuild.2018.03.020