Abstract
We present “Hive”, a software plug-in in Rhino Grasshopper for teaching integrated building energy systems design. The purpose of Hive is to provide an easily comprehensible tool for architecture students to establish an understanding of the impact of their design decisions on building performance and (renewable) energy systems. We enable a flexible framework allowing coupling with other third party (Grasshopper) components by proposing a so-called simulation core, which decouples calculation modules from the input-output interface. We test Hive with architecture students and present results from a user survey conducted. Our findings show that students find the climate and daylight analysis tools of Ladybug and Honeybee slightly more graspable for their design work, when compared to the energy tools in Hive – presumably also due to building energy being inherently less intuitive than daylight and climate analysis. The survey further shows that visual representation and ease-of-use appear to have priority over simulation accuracy and calculation time. However, we are convinced that our first version of Hive is a good starting point for accessible consideration of renewable energy systems into architectural design, fostering interdisciplinary collaboration.

Key Innovations
- Didactic software tool for integrated building energy (systems) design
- Simple input-output user interface for low entrance barriers
- (Partially) Real-time impact visualization of design choices on building performance
- Flexible framework to allow coupling with 3rd party libraries
- Student surveys to identify gaps and barriers of energy driven building design tools

Practical Implications
Hive facilitates a broader understanding of early design stage decisions, which is crucial for energy-positive and zero emission buildings. In addition, Hive provides selected but meaningful metrics and visualizations to foster a common language between all stakeholders and to boost articulation and implementation of sustainable building concepts.

Introduction
Synergistic effects between the energy demand and supply side suggest that building energy systems should be treated as integral parts in the design of buildings (Ferrara et al., 2019; Schlueter and Theseling, 2009). In architectural practice and education, however, building systems seem to be commonly regarded as to be specified by (HVAC) engineers only after the architectural concept has been drafted. While a separation of (technical) expertise appears indispensable given an ever increasing complexity of building projects, energy related considerations should nevertheless influence architectural design decisions at an early-stage (Lucon et al., 2014). Computational tools are one option for ensuring an effective knowledge transfer.

The existing literature identifies ease-of-use, interpretability and fast computation as most important characteristics for decision support tools in sustainable building design and optimization (Attia et al., 2013; Shi et al., 2016; Cichocka et al., 2017; Nault et al., 2018; Basic et al., 2019). While there has been an impressive ongoing surge of new tools being developed and released in the past years, there appears to be a lack in tools specifically addressing early stage integration of energy systems into building design. Hence, we propose a tool that aims to foster an understanding of the interplay between (renewable) energy systems and architectural design parameters. This can serve as a platform and common language for further collaboration between architectural designers and technical specialists.

In the following, we introduce our tool, including its distinctive characteristics and a proposed software framework that allows easy extension with third party libraries. Furthermore, we present survey results conducted with architecture students, identifying current shortcomings of, and potential improvements to the
The plug-in is open source and can be accessed online, can also be (double) curved.

Surfaces for solar energy systems (PV and ST), how-ever, can also be (double) curved.

The tool supports flexible geometry modeling to allow for volumetric architectural studies, including windows and surfaces for solar energy technologies, low for volumetric architectural studies, including windows and surfaces for solar energy technologies, as this eases coupling to common dynamic thermal systems, and the significant role that the environment (e.g. climate and urban context) plays thereby (Fig. 1).

The most essential energy and green house gas (GHG) related aspects are covered in the tool: shading mask analysis, simulation of building heating, cooling and electricity loads, solar irradiation on transparent surfaces, solar energy generation of photovoltaic (PV) and solar thermal systems, and basic embodied and operational carbon emissions, as well as investment and operation cost. Details on the simulation capabilities will be described in a later section.

The inputs are divided into three types: “Environment”, “Building” and “Energy Systems”. The environment is defined by surrounding geometries (neighoring buildings, trees, topography, etc.) and a weather file. The building inputs include geometries, thermal zone specifications (internal gains, air flow rates, etc.) and construction details (U-values, cost, embodied emissions, etc.). By default, thermal zone specifications are lumped together by “use-types” (residential, office, school, etc.), but individual parameters can be overwritten manually. Energy system inputs include energy carriers (grid electricity, natural gas, etc.), selection of energy conversion technologies (PV, heatpumps, etc.) and heat/cooling emitters and their supply temperatures (underfloor heating, air diffuers, etc.).

Geometric information can be directly connected into these three input components, other specifications can be set via a graphical user interface (GUI), or by connecting special parametric input components. Latter ease the workflow for sensitivity analysis or optimization. The GUIs can be accessed with a double-click on the input Grasshopper components, upon which a form is opened.

These three input types (environment, building and energy systems) are connected into a central “simulation core”, which computes output quantities that are visualized in 3D in the Rhino viewport, as well as on a results visualizer component in Grasshopper. The results visualizer serves as a GUI to navigate through different performance metrics and simulation results directly in Grasshopper. Current plots included are a Sankey diagram showing losses and gains into the thermal zone, solar irradiation on each window, solar energy generation, and building thermal and electric loads. Key performance indicators for cost, carbon emissions and energy consumption (investment and operational) are also listed. Fig. 4 shows two plot examples of the results visualizer in Grasshopper.

The major difference of Hive to most other tools is the integration of energy systems (generation / con-

---

**Hive – Tool introduction**

Hive is a plug-in for energy integrated building design, implemented in Rhinoceros 3D Grasshopper. The goal of Hive is to provide an easily comprehensible tool for architecture students to learn about the relation of their design proposals to building performance and (renewable) energy systems, and the significant role that the environment (e.g. climate and urban context) plays thereby (Fig. 1).

Distinctive features of Hive

**Hive**, as a didactic tool, is specifically tailored to teaching with the overarching goal that students learn to understand performative consequences of their design decisions on energy efficiency, comfort, cost and emissions, without needing to bother with the many technicalities and details that come with fully fledged simulation programs. Consequently, the main development goal was to achieve a framework that allows designers to focus on the inputs (building design) and how their decisions relate to performance outputs (e.g. energy, cost and emissions), while “hiding” calculation and simulation details from the user. Fig. 2 shows the general tool concept and Fig. 3 its realization in Rhino Grasshopper.

---

1. https://www.rhino3d.com
2. https://github.com/architecture-building-systems/hive. Release 1.7.0 is used in this paper.
version technologies and emitters) into building design at an early stage. The overall teaching objective follows a (recursive) process chain: (i) analysing environmental and climatic characteristics, (ii) identifying on site renewable energy potentials, (iii) utilizing passive design principles to minimize building loads, and finally (iv) selecting and sizing appropriate active energy systems. Having a simplistic, yet sufficiently representative tool at hand for this chain of tasks facilitates an understanding and collaboration between disciplines.

Other objectives in tool development were fast simulation times and comprehensible visualizations to allow for an intuitive learning experience, a simple user interface with low entrance barriers, and a software architecture that promotes open collaboration and extensions using third party plug-ins.

**Software framework – Hive.IO**

Hive uses a flexible software architecture that facilitates coupling to third party model libraries and other Grasshopper plug-ins. It is realized by establishing simple input-output relations, as well as by the introduction of simulation “distributor” and “results” merger components that handle data flow into and out from the simulation core (Fig. 5). These components belong to the project Hive.IO and they contain, in principle, no simulation models themselves.

There are three types of Hive input classes: Hive.Environment, Hive.Building and Hive.EnergySystems. The initialization of these classes is invoked automatically when the user connects Rhino geometry and when specifying other variables (weather file, energy systems, construction properties, etc.) in the input components.

Objects of these three classes are connected into the same node of the central simulation core (Fig. 3). The data flow starts with a main distributor component, which again exposes the three Hive classes separately. While this might appear as a redundant step, it allows the user to simply connect inputs into this simulation core without having to deal with the internals of the core.

Sub-distributors for different simulation / calculation modules fetch any required Hive object from the main distributor and further process the inputs into appropriate data types for their respective models (demand calculation, solar simulation, energy systems, etc.).

As Fig. 5 shows, results of all simulation models (grey boxes) are collided in an energy systems component (blue box). This will be further elaborated in the next section on the simulation specifics.

At the end of the simulation core, all computed results are streamed together in a central results merger that processes this data into a Hive.Results object. Finally, this object contains all results that can be displayed in the results visualizer (Fig. 4).

Using this framework architecture, new simulation modules can easily added by writing a new sub-distributor based on the Hive API. Possible third-party extensions could be for more detailed life cycle analysis using Bombyx (Basic et al., 2019), or daylight...
and climate analysis using Honeybee and Ladybug (Roudsari and Pak, 2013).

Simulation core – Hive.Core

The hierarchy of models in the simulation core (Fig. 5) is based on an energy balance:

$$L = CP,$$

s.t.  $P_{lb} \leq P \leq P_{ub},$

where $L$ are building loads, $C$ represents an energy conversion matrix and $P$ is the power flow (heat, electric, solar, etc.) bound by upper and lower limits $P_{lb}$ and $P_{ub}$ (e.g. solar potentials or technical constraints of energy systems). When computing operational emissions and costs of a building energy system, building loads and power flow bounds need to be known beforehand. The grey boxes in Fig. 5 represent such simulators. Consequently, components that concern energy conversion and generation systems are at the end of the simulation core, since $L$ and $P$ are necessary inputs to this system (see second most right blue box “Energy Systems” in Fig. 5). The formulation of the simulation core follows that of the energy hub concept, which allows for operational and design optimization of more complex multi-energy systems (Geidl and Andersson, 2007), and we plan to add this to future releases.

Furthermore, to remain consistent with the software architecture outlined in the previous section, all simulation models used in the current build are separated from the Hive.IO project and instead belong to Hive.Core. These models are also Grasshopper components, but can in principle be used without Hive.IO. The simulation models are written using a custom open-source Grasshopper compiler for Python, called Honeybadger$^3$.

Demand calculation:

Heating, cooling and electricity loads are calculated based on SIA380 (2009) for a monthly resolution, but we intend to extend it to an hourly resolution in the future. Even though it is a static model, a coefficient ($\tau$) that is based on the building construction emulates thermal latency. Adaptive comfort ranges are calculated based on Auliciens and Szokolay (2007). Our motivation for choosing such a demand modeling approach is to minimize simulation times: this model solves instantly on an Intel i7-8700K processor.

Building energy systems:

We have implemented either simplified equations for calculating efficiencies and coefficients of performance (COP) for air-source heat pumps, chillers, boilers, solar thermal and PV systems, or time resolved equations from Ashouri et al. (2013); Choi et al. (2005); Mavromatidis et al. (2015); Omu et al. (2016). The calculation method can be set by the user. Currently, an air-source heat pump is set by default and its COP depends on the ambient air temperature and the supply temperature. Electricity is provided by the grid. Energy generated from PV and solar thermal collectors is self-consumed, excess electricity sold to the grid and excess thermal energy discarded. Thermal and electric storages are not implemented yet, but given their importance we intend to include them in the future.

Solar simulation:

As mentioned earlier, the simulation core is designed to be compatible with third-party plug-ins. As a proof of concept, the current build of Hive uses an external Grasshopper plug-in$^4$ for simulating solar irradiation and real-time shading masks for obstruction analysis (Fig. 6). The model is validated against Radiance, Daysim and EnergyPlus in Waihel et al. (2017). Currently, the annual hourly solar simulation is the bottleneck regarding computation time. Given the flexible framework of Hive, it could be easily replaced with faster programs using cloud computing, graphics processing units (GPU) or machine learning.

Hive database

We have implemented a basic input dataset using the Swiss context, due to the tool being developed and used in Switzerland. Database values can be overwritten either manually via the input GUI, or extended by using the Hive API and following our json naming conventions.

The database is structured into five main categories: Thermal building envelope, use-type, technical performance of building systems, embodied and operational emissions of energy systems and materials, and operational and investment cost. Relevant weather
data is extracted from any epw file specified by the user. Sources used are summarized in Table 1.

It should be critically mentioned that much of the data used is highly fluctuating, uncertain, and/or is subject to a continued change. This becomes especially apparent when examining the price evolution of PV within the last years. Data bases should therefore be constantly updated.

**Thermal envelope properties:**

Thermal properties according to the building energy demand calculations from SIA (Swiss Society of Engineers and Architects) 380/1 are used (SIA380, 2009). Typical values for the existing building stock, standard quality and target values are given in SIA 2024 and implemented respectively (SIA2024, 2006).

**Use-type:**

The use-type influences set points, schedules, internal gains and ventilation / infiltration rates. Standard values are found directly in SIA380 (2009) or in SIA2024 (2006). Respective values are parsed by choosing the use-type via the GUI (e.g. multifamily house, office, etc.). Parameters can also be overwritten manually in the GUI.

**Technical performance of building systems:**

Several models for building system components are used to calculate photovoltaic and solar thermal yield, the seasonal coefficient of performance (SCOP) of heat pumps and efficiencies of combustion based heaters. Typical PV technology efficiencies and performance ratios are implemented with data from state-of-the art literature (Philipps and Warmuth, 2020). The heat pump performance is based on physical equations relating to the outside temperature. Combustion based systems have constant efficiencies. Again, parameters can be set via the GUI.

**Life cycle impact:**

We provide estimates of greenhouse gas emissions for assessing the life cycle impact. These are split into embodied and operational emissions. The data is taken from the Swiss database of construction materials KBOB 2016 which also provides emission values for energy sources (Die Koordinationskonferenz der Bau und Liegenschaftsorgane der Öffentlichen Bauherren (KBOB), 2016). Further, the Swiss cost catalogue for building elements also provides embodied emissions for envelope cross sections (CRB, 2011). For different PV technologies that are not yet part of KBOB, additional sources are used (Frischknecht et al., 2014).

**Cost data:**

For the Swiss context, construction cost estimates for typical envelope structures are available from the Swiss cost catalogue for building elements (CRB, 2011). The cost values are linked to the envelope quality chosen above, but, again, can be overwritten manually. Several sources online and from the literature have been used for energy systems cost data. Latest online sources are used for PV cost and efficiency parameters (Philipps and Warmuth, 2020).

**User survey**

In autumn semester 2020, we have used Hive in our university course with 51 Master students. The students’ backgrounds were mostly in architecture with few from engineering departments. In this section we present survey results that we have conducted at the end of the term. The course deals with passive and active design strategies and methods to design buildings that respond to local climate. The course puts a significant emphasis on design and analysis exercises using Rhino Grasshopper, including climate analysis and daylight tools from Ladybug and Honeybee. From the survey conducted, we can identify qualities and challenges of the current tool version, as well as compare it to the other tools used.

It should be mentioned that almost none of the students had experience in Grasshopper and only few were accustomed to Rhino. Therefore, these skills had to be acquired along the course, making the survey a good precedent case for easy-to-learn performance and sustainability tools.

The survey was conducted via an online form and
contained mostly likert scale matrix questions about the three tools Hive, Ladybug and Honeybee. Out of the 51 students, we received 19 complete survey responses. The low response rate may be explained by the time that the survey was conducted (before new year, and a reminder in early January 2021). Furthermore, the students had to fill out course evaluations at the same time, which might explain a certain exhaustion and lower willingness of filling out more surveys. Nevertheless, we believe that we can gain valuable information from the respondents.

We omit statistical significance tests in the following analysis, mainly due to the very small sample size (19), but also because the tools (Ladybug, Honeybee and Hive) are in fact difficult to compare due to their different scopes. We therefore interpret the results based on colloquial feedback of the students and observations in the course. Our assumption is that if students perceive energy topics similarly or only slightly less easy to learn as climate analysis, the tool was successful.

**Importance of tool features**

Fig. 7 shows students’ ratings on the importance of various tool features. The size of a circle corresponds to the counts per answer. The exact question was: “Which tool characteristics are important to you and would make you use such a tool in your design project?”. The rows in the matrix were:

- Quickly learn the simulation environment.
- Fast computation of the results.
- Simple interface to use.
- High accuracy of the results.
- Attractive visual representations of results.

Possible answers were “Not at all important”, “Not so important”, “Neutral”, “Important”, “Very important”.

At first sight, all asked features appear to be important to the students, which might be due to the question being a likert scale matrix; a ranking question might have yielded in a clearer picture. Looking at it more closely indicates a high preference of being able to learn the tool quickly. One possible interpretation might be that nowadays, new tools emerge rather frequently and required competencies in a profession change faster than ever.

In such an environment, easily accessible and comprehensible tools might significantly increase the efficiency of education and decision making. The worst case situation would be a long training phase for a specialist tool which becomes obsolete within a short time already. Considering the current technological transformation of the energy landscape (e.g. renewables, smart cities and buildings) (UNEP, 2015), it is very likely that (architectural) decision making tools continue to develop.

Computation time seemed to be least important. It should be mentioned, however, that the used models in Hive complete within seconds or few minutes (depending on hardware and mesh resolution for the solar simulation). As most students attending this course did not have prior experience in building simulation, it might well be that they are not aware of computationally more expensive simulations and they take the fast simulations exposed to them in this course as normal.

The questions (abbreviated) “Simple UI” and “Attractive results” rated similar as second most features, after “Learn quickly”. These two questions have consistently been rated similar to each other throughout the survey.

As for the “Accuracy” of the simulations, the majority rated it as “Important”, which suggests that students expect models to be representative enough for making relevant design decisions, but can tolerate a deviation from fully fledged expert programs.

**Comparison of tools**

Fig. 8 may indicate the usability of different tools and applications to architecture students and it shows their responses to the questions

- “I feel confident to use this <tool> in a design project in the correct modeling situation.”
- “I am aware of the limitations and scope of the <tool>.”

<tool> included Ladybug for climate analysis, Honeybee for daylight simulations and Hive for energy analysis. The ratings were “Strongly disagree”, “Disagree”, “Neutral”, “Agree”, “Strongly Agree”.

The results show that students are highly confident in using Ladybug for climate analysis, closely followed by Honeybee for daylight analysis. The rating for Hive for energy analysis is lower, with the median being at “Neutral”. The ratings for the question regarding limitation awareness look similar.

This might be explained by the fact that climate analysis and daylighting are very visual topics that might be more intuitive to grasp for architectural designers in comparison to energy (systems). It also shows that in order to truly integrate energy into architectural
building design, tools need to be highly accessible and intuitive.

Improving Hive

Fig. 9 shows responses to the likert scale matrix questions:

- “How do the characteristics below apply to the HIVE templates that were presented in the course exercises?”
- “Which characteristics should be further improved to make you frequently use a Grasshopper tool (HIVE) in your design projects?”

The rows were the same as for the question regarding the importance of tool features, but we omit responses for “Attractive results”, because it has similar responses as “Simple UI”.

The response option for the first matrix were the same as for the matrix on the importance of tool features (“Strongly disagree” to “Strongly agree”). The response options for the second matrix were: “Needs significant improvement”, “Needs many improvements”, “Needs few improvements”, “It is OK after minimum additions”, and “Perfect as it is.”

Interestingly, the students responded that Hive succeeds to a large extent for all features (blue boxes in Fig. 9) with the median and/or the upper quartile being at “Agree” (one tick below “Yes” in Fig. 9).

When it comes to which features should be improved (red boxes in Fig. 9), again, students seemed rather content with the tool. Least improvements seem to be requested regarding computation times, which appear to be right for the students. Slightly more students request “accuracy” and “being able to learn the tool quickly” to be improved.° The user interface (UI), however, seems to be the most requested feature to be improved. This would include visualizations and the GUIs. We hypothesize that a better UI and more

Figure 8: Boxplots with student responses to comparing tools. “Yes” corresponds to “Fully agree” and “No” corresponds to “Fully disagree”. Notches indicate the medians; triangles are graphical indicators that the respective quartile is condensed in the median.

Figure 9: Boxplots of student responses to the question where Hive succeeds and where it needs to be improved. Y-axis labels “Yes” and “No” correspond to the blue boxes, “can stay” and “needs impr.” correspond to the red boxes. Notches indicate the medians; triangles are graphical indicators that the respective quartile is condensed in the median.

(3D) visualizations would also improve the confidence in using the tool, as well as in facilitating an understanding of energy systems integration.

Summary and conclusion

This paper presented our first official release of Hive, a tool for energy integrated design in teaching and early stage decision making. Distinct features are its tight integration with energy systems, particularly solar technologies, and its flexible software architecture allowing easy extensibility with third party plug-ins.

A user survey with architecture students has shown that the tool is successful in providing an accessible and comprehensible platform for making energy driven design decisions, albeit it minimal prior knowledge in the software (Rhino Grasshopper) and no experience in building simulation. When compared to climate and daylight analysis tools from Ladybug and Honeybee, however, the energy tools of Hive seem to be more difficult to apply.

To foster the uptake of such tools, we found that hiding simulation specifics from the user into a simulation core and mainly exposing the user to input-output relations proved effective. This includes providing robust default values, such as for simulation model settings or technology parameters.

Furthermore, the survey showed a clear preference towards tools that are easy and quick to learn. User interfaces and results visualizations can and should further be improved, as this will ease an intuitive in-

°Students’ ratings to “Accuracy” may be interpreted as how much they believe in the tools’ calculations. As a requirement to participate in this course, the students were expected to have completed previous courses on building physics and energy systems, where they have learnt most of the calculation principles and steady-state equations as implemented in Hive.
tegration of energy related information into building design.

As a final conclusion, we believe that in light of the growing complexity of building projects and an increasing interwovenness of specialist disciplines, it becomes indispensable to offer a design frameworks that allows for a holistic yet intuitive workflow. This way, a common language across domains can be established, further intensifying the necessary exchange between high level designers and domain experts.

Acknowledgment

This work has been supported by an ETH Zurich Innovedum grant. Furthermore, we would like to thank the various contributors to Hive, as well as the ETHZ students of Energy- and Climate Systems 3, autumn semester 2020, for participating at the survey.

References


