

Free and Open-Source Teaching: Understanding Exergy using Thermal Engineering Systems in Python (TESPy)

**Mathias Hofmann^a, Francesco Witte^b, Malte Fritz^{c,d}, Jonas Freißmann^{c,d},
Ilja Tuschy^{c,d}, George Tsatsaronis^e**

^a Technische Universität Berlin, Berlin, Germany, hofmann@iet.tu-berlin.de, CA

^b German Aerospace Center (DLR), Institute of Networked Energy Systems, Oldenburg, Germany

^c Hochschule Flensburg, University of Applied Sciences, Flensburg, Germany

^d Center for Sustainable Energy Systems (ZNES), Europa-Universität & Hochschule Flensburg, Flensburg, Germany

^e Technische Universität Berlin, Berlin, Germany

Abstract:

In energy or chemical engineering, a representation of highly complex processes is often only possible through a simulation. Conducting experiments or building demonstrators is far too costly and time-consuming. Applying exergy-based methods is beneficial to reveal thermodynamic inefficiencies and to propose appropriate optimization approaches for energy conversion processes. Therefore, teaching theory and software-based application of such methods are essential in engineering study programs.

This paper presents a didactic concept for open educational resources on exergy analysis covering the introduction to the exergy method, its distinction from the energy analysis, and its application to various case studies. The course aims to combine thermodynamic understanding with numerical mathematics and object-oriented programming. Students learn to build and run models of energy conversion systems and conduct respective exergy analyses using the software Thermal Engineering Systems in Python (TESPy). The course material is developed using Jupyter notebooks, which offer a flexible connection between theory and code. Results are reproducible and can be tested and developed further by the open-source community.

The advantages of exergy-based methods can be illustrated by evaluating and visualizing real thermodynamic losses using Grassmann or waterfall diagrams. Process understanding can be deepened further by parameter analysis showing their impact on components and the overall process. Finally, the acquired knowledge is transferred to more complex problems with multiple components or more than one product.

Keywords:

Thermodynamics, Exergy, Exergy-based methods, Simulation, Python, Online class

1. Introduction and Motivation

In the last years, particularly since the start of the COVID-19 pandemic, the need for using electronically-based teaching materials has increased. Nevertheless, the didactic concepts for teaching events have also changed fundamentally in a concise time [1, 2]. Specific methods, e.g., flipped/inverted classroom, were previously only used by an interested group of lecturers and played a subordinate role. The pandemic has significantly accelerated the development and led to the increased use of numerous previously underrepresented didactic methods. As a result, the didactic quality of teaching courses was improved almost imperceptibly, especially in the field of engineering.

But what principles should be applied to the design of a lecture or seminar at a university today? Von Thun [3] gives some advice on this. He considers the development of one's own design principles as fundamental, which correspond to the requirements and possibilities of your subject, your contents, and your person. The lecture or seminar should not only be given but also created. Developing those design principles is an exciting and valuable part of the professional biography of a university teacher.

When designing a course, it is necessary to determine which competency goals are to be achieved. According to Anderson and Krathwohl [4] there are six categories of cognitive process dimensions to choose from: first remember, second understand, third apply, fourth analyze, fifth evaluate, sixth create. Many courses at universities, especially introductory classes, usually are within the third category. However, the study's goal should be to reach the 6th level, i.e., to plan a research paper on a given topic, i.e., to work scientifically on one's own. Concerning engineering sciences, it is, therefore, essential to get into independent practice-relevant action as early as possible in the curriculum, e.g., to be able to program independently.

In engineering programs, especially in the field of energy and chemical engineering, thermodynamics is a fundamental discipline. In many of these courses, the exergy concept [5–9] is taught to aspiring engineers as a helpful and powerful tool. In this context, the conclusive teaching of the concept is one of the major challenges for the lecturers. There needs to be more than a conventional thermodynamics lecture to make the students apply the exergy concept and explore exergy. Following the previous remarks, we need at least one more advanced course for teaching the exergy concept. Independent activities such as programming should be included. The students must be enabled to apply general principles of thermodynamics independently, evaluate them, and finally, create energy conversion processes independently according to the given requirements and check them with the help of exergy-based methods.

In today's highly interconnected and globalized world, it should be widely accepted that research and teaching are not the exclusive preserve of established societies or segregated groups with limited social access. Indeed, even in the beginnings of the universities in the Middle Ages, admission restrictions were unusual [10, 11]. Ultimately, the composition of the student body was shaped by social realities, such as the exclusion of women. Of course, numerous social barriers have been dismantled in the meantime. That it is nevertheless not self-evident is shown by the 4th United Nations Sustainable Development Goal¹: “Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.” Addressing lifelong learning shows a need to make knowledge and knowledge transfer accessible to all without barriers, even outside appropriate institutions. For example, this can be made possible through a digital offering independent of end devices. Free and open-source access should be a standard feature.

Furthermore, in research projects funded by third parties, work is often expected today to be transdisciplinary. Consequently, social shareholders are also involved, and data and results should be available to everyone for subsequent use. For such cooperative collaboration across spatial or temporal distances, several technical solutions (cloud, git, web) exist today that enable participation and curation in various forms simultaneously. Remember, understand, apply, analyze, evaluate, and create no longer necessarily have to take place in the lecture hall or seminar room on the chalkboard.

An online course was developed based on the findings formulated up to this point. The open educational resources on exergy analysis covering the introduction to the exergy method, its distinction from the energy analysis, and its application to various case studies. The interactive digital learning experience offered to students in the course described here is a building block in the development of professional competencies for future engineers.

The present contribution is structured as follows: A didactic concept for the developed online course is presented in the next section. Information about the online implementation based on Python and Jupyter Notebooks is collected in Section 3. Some illustrative results from the course material are shown in the results section, followed by the conclusions.

All developed materials and used libraries are available on a free and open-source basis. If researchers, practitioners, teachers, or students use the content, the CC BY 4.0 guidelines² should be respected. The prepared content can be accessed online in a textbook format.³ Furthermore, everyone is invited to join the developer team, share ideas and suggestions, or even bug reports via the GitHub repository.⁴

2. Didactic Concept

2.1. Target Group, Prerequisites and Integration into Curriculum

The course is intended for students in undergraduate engineering programs, especially power engineering, process engineering, and chemical engineering. In addition, the course can be a possible advanced training, since it is available in an open-source form. Nevertheless, the content is not primarily aimed at that, meaning that examples could seem too simple for a practitioner.

Students should already completed coursework in basic mathematics, thermodynamics, and object-oriented programming knowledge. Furthermore, they should be familiar with the purpose of using apparatus and machines as well as processes of energy engineering. Finally, they are expected to be interested in going deeper into the exergy concept.

Consequently, the course fits into the curriculum of the study program according to Figure 1. In deviation from the illustration, it is conceivable that the course is offered in the third or fourth semester. In any case, the stated prerequisites should be taken into account. The advantage of an early start of the course is the practice-oriented relevance of the contents. Thus, a hands-on mentality is developed, and the students are taught the relevant engineering skills at an early stage. They learn to solve problems in teams and practice modeling, programming,

¹<https://sdgs.un.org/goals/goal4>

²<https://creativecommons.org/licenses/by/4.0/>

³https://fwitte.github.io/TESPy_teaching_exergy/

⁴https://github.com/fwitte/TESPy_teaching_exergy

and scientific computing. Also, they get familiar with from scratch process design and find alternative solutions. Finally, the students further develop existing structures.

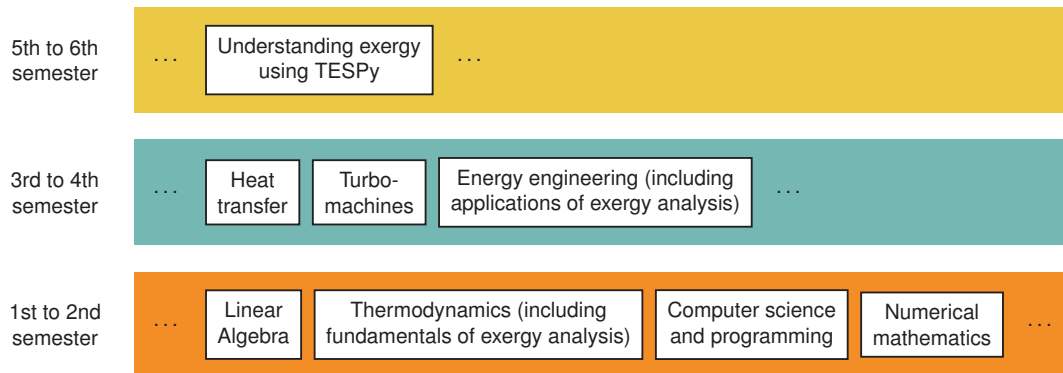


Figure 1: Integration into the curriculum of a typical engineering undergraduate program

2.2. Learning Outcomes

After finishing the course, the students are familiar with the basics of steady-state process simulation and know different methods for calculating the thermodynamic properties of substances. Also, they can build, initialize and solve process flowsheets independently, using the software Thermal Engineering Systems in Python (TESPy) as a process simulation tool for modeling, calculating, and analyzing complex energy engineering processes and interpreting the results obtained. The participants identify and solve problems that occur during process simulation and can independently use aspects of the exergy concept for their future research questions. They are familiar with physical and chemical exergy and the splitting of physical exergy. They can evaluate processes and components with the help of exergy analysis and exergy-based parameters and present the analyses graphically. Finally, using a version control system, the students apply group work methods such as team communication, milestone processing, and cooperative programming.

In their best efforts, students complete the module by adding new elements to the existing course, and thus reach the highest level of competence according to Anderson and Krathwohl [4], see Section 1.

2.3. Content and Competencies

Following the course content of fundamentals of thermodynamics and energy conversion, the focus is on simulating single state changes, simple cycles, and complex energy conversion systems. At the same time, the participants will practice methods of exergetic process evaluation.

The mathematical representation of individual components of energy engineering processes, e.g., heat exchangers, pumps, turbines, waste heat boilers, steam generators, condensers, is subsequently combined into the representation of overall processes, e.g., gas turbines, gas and steam turbine power plants, steam power processes, and integration of renewable energy sources. The methods of setting up process simulations, solving mass and energy balances, and calculating thermodynamic property data are covered. The simulation program used is the free and open-source software TESPy, see Section 3.

The work can be done individually or in small groups. We recommend working in teams of 2 to 3 people to intensify social skills and test suitable collaboration methods. Figure 2 shows the distribution of the competence fields addressed during the course. Depending on the individual characteristics of the course, shares may shift slightly.

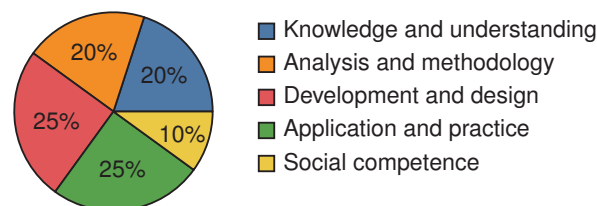


Figure 2: Pie chart showing the competency fields addressed in the progress of the course.

2.4. Schedule

The Gantt chart in Figure 3 shows a sample schedule for designing the class for one semester (2 contact hours per week). In its first version, the course includes four sections and three milestones. The kick-off meeting can include a get-to-know-you and orientation session if the students work in teams. Afterward, there is the possibility to configure the own hardware and run tests regarding the functionality of the libraries to be used.

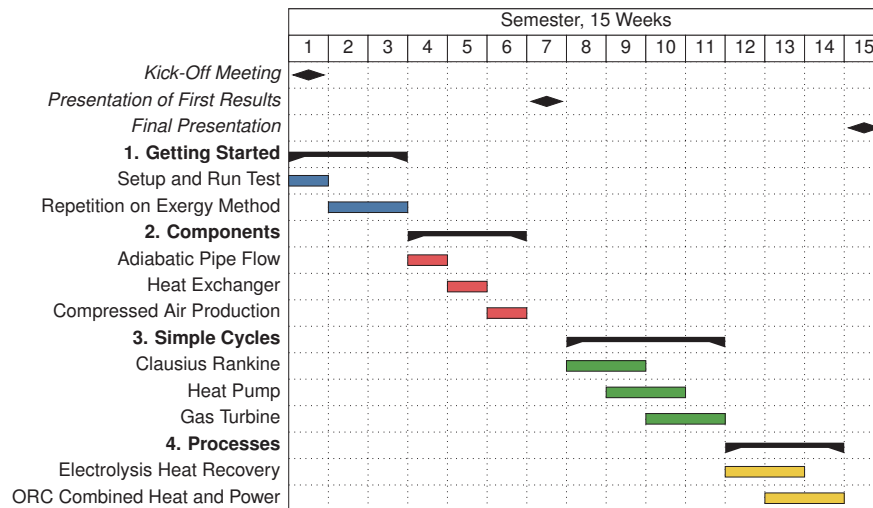


Figure 3: Gantt chart showing the possible progress of the course in one term.

In the second section, students are encouraged to investigate thermodynamic fundamentals for simple applications and program them independently. All analyses are carried out at the component level and usually involve exactly one thermodynamically describable change of state. With the help of simple exergetic analyses based on the solution of the mass and energy balances and simple graphical evaluations, differences between energetic and exergetic descriptions can be made easily understandable.

In the second section, the students will partially write down the correlations in an elementary way and later moving forward to a object-oriented programming. The advantages of a Python library (e.g., TESPpy) become clear with the transition from the second to the third section.

In the third section of the course, the exergy analysis is to be applied to simple multi-component systems. At the latest in this section, the students should be aware of the advantages of the exergy-based evaluation compared to a purely energetic consideration. A further focus is then on the graphical evaluation of the results. How can the findings be visualized in the best possible way? The students can try out various approaches independently here.

Since the interaction of several components is now being investigated, the focus is on how the components' properties impact the overall process. For instance, which components significantly influence the exergy destruction of the overall process and why. It is necessary to determine which thermodynamic parameters are the most important for improving the overall process.

The last section examines complex processes with more than one product. If a process provides different forms of energy as products, for example electricity and district heating, an energy-based assessment of the quality of the overall process is misleading. Exergy is the only way to ensure comparability of the different material and energy flows. The students are to conduct these evaluations for processes such as heat recovery in electrolysis or processes with combined heat and power.

2.5. Further Reading, Online Tutorials, Hardware, Software

During their work the students can use several textbooks or online tutorials for fundamental or further reading. We recommend the following:

- Thermodynamic fundamentals [12, 13]
- Exergy concept [6–9]
- Python [14] and Jupyter Notebooks [15]
- TESPpy [16–18]

For scientific computing in Python environments, it is a convenient option to use ready-made distributions that integrate relevant packages and library management in a user-friendly way. A simple start succeeds with miniforge⁵, available for all current operating systems. To edit and run the code students can install a code editor such as Pycharm⁶ (free of charge for university members) or Visual Studio Code⁷ or run a local server allowing them to work in a browser. The online documentation of the course includes all relevant installation instructions.

3. Implementation

In recent years, there has been a concern [19] in the energy system analysis community, among others, with the public provision of data and publicly funded research's traceability. Various initiatives are engaging in this area, developing free and open-source work paradigms and taking them into other domains. Aspects initially discussed only in research are spilled into teaching. For example, the Journal of Open Source Education [20] specifically addresses lecturers to publish online educational resources connected with free and open-source software. In a proposal, Morrison [21] formulates the idea of combining publicly available data and voluntary contributions from experts into a "student-centered" energy system model for education. The actual energy system model will be programmed in Julia, and the workflow will be set up in Jupyter. It is explicitly not a web-based scenario generator or gamification exercise. The students will perform exercises in a curated way and develop the code of the energy system model piece by piece.

Inspired by these ideas, the present course for energy technology and exergy analysis was developed. In particular, the idea for the structural content and the technical project architecture is based on Morrison's proposal. The entire content of the course is embedded in a Python-based structure. The prepared content can be accessed online in a textbook format⁸ and the source files are available through the respective GitHub repository.⁹ Figure 4 shows the implementation.

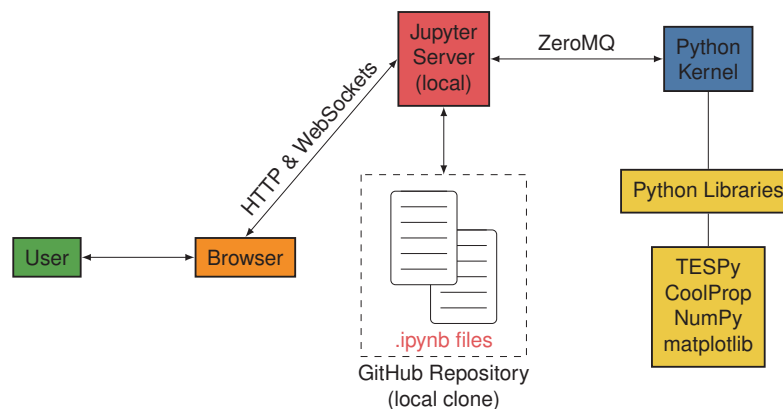


Figure 4: Implementation of the resources into a Python-based architecture including a powerful and user-friendly front-end

The files¹⁰ are provided as a Jupyter notebook. It is an editable format where not only Python code is stored and can be executed, but visualization, markdown, and collaboration capabilities exist. Consequently, this format is particularly suitable for course formats like the one presented here. Outlines can be created, and text can be formatted, images and videos can be added, mathematical formulas and referencing can be used.

To create their own code, students can download the Jupyter files to their machine and locally run a Jupyter server, which allows to access a Python kernel and run their code in the browser or in code editors.

The open-source software TESPpy [16] is used for the component-based simulation of the considered energy conversion processes. The software is developed as part of the open energy modeling framework (oemof) in the scientific field and by industrial users.¹¹ The user can build up arbitrary thermodynamic processes from single predefined components like heat exchangers, compressors, valves, or turbines. In addition to heat pumps, geothermal or solar thermal power plants, heat networks, or chillers can also be mapped. The software

⁵<https://github.com/conda-forge/miniforge>

⁶<https://www.jetbrains.com/community/education/#students>

⁷<https://code.visualstudio.com/>

⁸https://fwitte.github.io/TESPpy_teaching_exergy

⁹https://github.com/fwitte/TESPpy_teaching_exergy

¹⁰file extension .ipynb

¹¹<https://oemof.org>

accesses the likewise open-source substance properties database CoolProp [22]. Detailed documentation with more examples and tutorials is available for the software.¹²

Following process simulation and design, exergy analysis can be used to identify sources of thermodynamic losses. The exergy analysis is performed automatically. Only the exergetic fuel, product, and loss of the overall process have to be passed to the tool [17, 18]. Furthermore, the simulation models can be combined with optimization methods [23].

4. Featured Course Material

The course is based on three relevant pillars whose approach enables change management and innovation processes in their respective disciplines or application areas. The three pillars are

- object-oriented programming with Python,
- steady-state power plant simulation with open-source software (TESPy, CoolProp), and
- thermodynamic evaluation using exergy-based methods.

As a consequence, the contents of the course lead to the conclusion that these three pillars can be interlinked with each other exceptionally well for prospective engineers to teach the exergy concept. We want to summarize this by presenting selected tasks from the course and illustrating the results comprehensibly.

Within all listed thermodynamic models below we assume steady-state processes, all components are adiabatic, there are no pressure losses in the heat exchangers, and changes in kinetic and potential energies can be neglected. Ambient conditions are set to $T_0 = 25\text{ °C}$ and $p_0 = 1.013\text{ bar}$. For the calculation of chemical exergies the model of Ahrendts [24] is used. The corresponding flow diagrams and parameters of the processes can be found in Figures 5, 8, 11, and 12 as well as Tables 1, 2, and 3. Due to space limitations, not all source codes are given below. All tasks and solutions can be obtained from the git repository.

4.1. Highlight 1: Adiabatic Pipe Flow

Consider a well insulated pipeline transporting a fluid, i.e., water steam or air. The state change from 1 to 2 because of friction can be described as an adiabatic throttling valve as shown in 5.

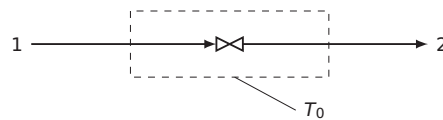


Figure 5: Flowchart of adiabatic pipe flow

Table 1: Adiabatic pipe flow parameters

Parameter	Symbol	Unit	Value
Inlet temperature	T_1	°C	195
Inlet pressure	p_1	bar	10
Constant (or varying) outlet pressure	p_2	bar	6 (or $p_1 \dots p_0$)
Mass flow rate	m_1	kg/s	1

First, setting up the parameters of the pipe flow and calculating the inlet enthalpy, if necessary install CoolProp.¹³

```
from CoolProp.CoolProp import PropsSI as PSI
fluid = "water"
p_in = 10 * 1e5
T_in = 195 + 273.15
h_in = PSI("H", "P", p_in, "T", T_in, fluid)
p_out = 6 * 1e5
```

Considering the above assumptions, and the energy balance, we determine the outlet temperature for isenthalpic throttling using CoolProp.

```
h_out = h_in
PSI("T", "P", p_out, "H", h_out, fluid) - 273.15
```

¹²<https://tespy.rtd.io>

¹³As an alternative a simple, full-featured, lightweight CoolProp wrapper for Python is available via <https://pypi.org/project/pyfluids/>.

Subsequently, the following exercises, among others, can be performed. Implementing a function that splits the physical exergy in its thermal and mechanical parts [25]. Creating a plot showing how the thermal and mechanical exergies are affected within a range from ambient pressure to inlet pressure. And re-performing the tasks by using air instead of water steam.

Using a Python function the calculation of thermal and mechanical exergy can be performed for different temperatures, pressures and fluids.

```
def calc_splitted_physical_exergy(p, h, p0, T0, fluid):
    r"""Calculate specific physical exergy according to splitting rule."""
    s = PSI("S", "P", p, "H", h, fluid)
    h_T0_p = PSI("H", "P", p, "T", T0, fluid)
    s_T0_p = PSI("S", "P", p, "T", T0, fluid)
    ex_therm = (h - h_T0_p) - T0 * (s - s_T0_p)
    h0 = PSI("H", "P", p0, "T", T0, fluid)
    s0 = PSI("S", "P", p0, "T", T0, fluid)
    ex_mech = (h_T0_p - h0) - T0 * (s_T0_p - s0)
    return ex_therm, ex_mech
```

```
ex_T_in, ex_M_in = calc_splitted_physical_exergy(p_in, h_in, p0, T0, "water")
```

Figure 6a and 6b show the difference in the values between inlet and outlet of the thermal and mechanical exergies, representing the corresponding exergy destruction in the pipe flow for varying pressure ratios. Note the different scales of the y-axes. The thermal exergy of water steam changes by up to 300 kW, while the mechanical exergy difference is 0.9 kW at maximum. For air the thermal exergy difference reaches a maximum of around 0.5 kW, the mechanical exergy changes up to 200 kW.

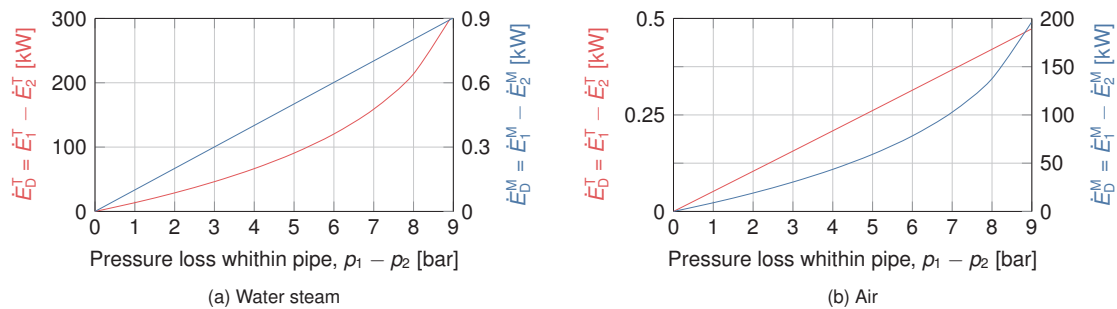


Figure 6: Thermal and mechanical exergy destruction within the the pipe for varying pressure losses

The fundamental difference of the substance characteristic leads to the fact that the exergy destruction for water steam in the pipe flow is mainly triggered by a decrease of the thermal exergy. In the case of air, the mechanical exergy decreases significantly more.

A comparison of the exergy destruction rate for the pipe flow of water steam and air is shown in Figure 7. The ratio of both exergy destruction rates is almost constant and is about 1.5.

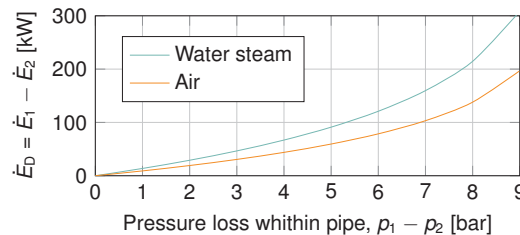


Figure 7: Exergy destruction for water steam and air within the adiabatic pipe flow

Lessons Learned from this task:

- Physical exergy of a mass flow can be split into a thermal and a mechanical part.
- At the same pressure and temperature, the shares of thermal and mechanical exergies depend on the working fluid.

4.2. Highlight 2: Open-Cycle Gas Turbine

A simple gas turbine model should be developed within this task. Compressed ambient air is mixed up with methane. The gases after the combustion are used in a gas turbine expander to supply mechanical, respectively electrical energy.

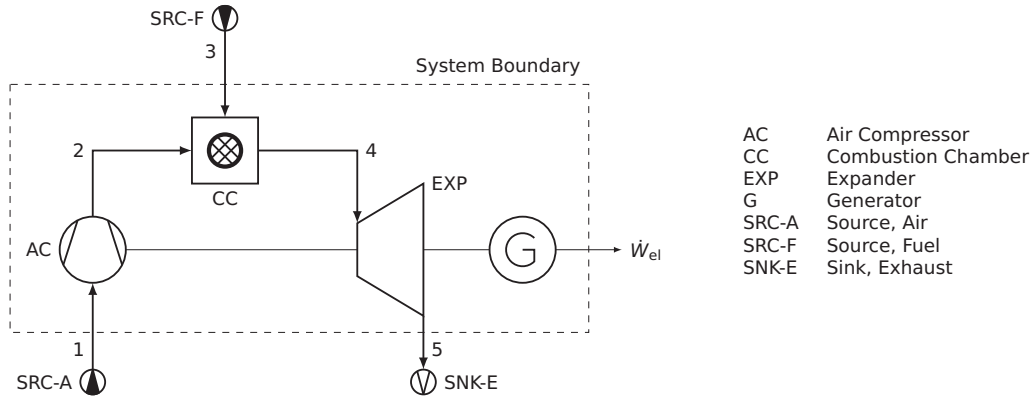


Figure 8: Flowchart of open-cycle gas turbine

Table 2: Open-cycle gas turbine parameters

Parameter	Symbol	Unit	Value
Net power output	\dot{W}_{net}	MW	150
Air compressor, isentropic efficiency	$\eta_{s,AC}$	–	0.85
Air compressor, pressure ratio	$r_{p,AC}$	–	18
Combustion chamber, thermal efficiency	η_{CC}	–	0.98
Combustion chamber, pressure ratio	$r_{p,CC}$	–	0.95
Expander, isentropic efficiency	$\eta_{s,EXP}$	–	0.9
Expander, inlet temperature	T_4	°C	1400
Expander, outlet pressure	p_5	bar	1.013
Fuel, inlet pressure	p_3	bar	20
Fuel, inlet temperature	T_3	°C	25
Fuel, mass analysis ^a	$y_{i,fuel}$	–	(..., 1.0)
Air, mass analysis ^a	$y_{i,air}$	–	$\begin{pmatrix} 0.7551 \\ 0.2314 \\ 0.0129 \\ 0.0006 \\ 0.0 \\ 0.0 \end{pmatrix}$

^a All fluid compositions are given as $y_i = (y_{N_2}, y_{O_2}, y_{Ar}, y_{CO_2}, y_{H_2O}, y_{CH_4})^T$

TESPy is used to set up the thermodynamic model of the gas turbine. The solution is carried out for the full load case. The documentation of the source code can be found in the appendix.

A parameter study should be performed based on the results for the air compressor pressure ratio and expander inlet temperature. The solution is a multiple run of the simulation above, while varying pressures and temperatures.

Figure 9 depicts a parameter plot for the given gas turbine system varying expander inlet temperature from 900 °C to 1400 °C and the air compressor pressure ratio from 5 to 30. The reference point corresponds to the data from Table 2. For the plot only the marked points were calculated, the lines in between are linear regressions. If necessary, the accuracy of the graph can be increased by performing more calculations. An optimal combination of expander inlet temperature and pressure ratio exists to maximize the specific work and the efficiency of the gas turbine.

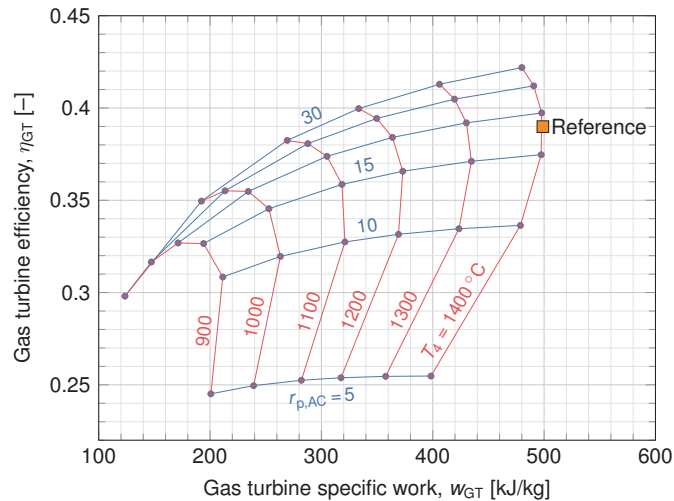


Figure 9: Gas turbine parameter plot for varying expander inlet temperature air compressor pressure ratio

However, with this initially only energetic evaluation of the process, based on the specific power and the net efficiency, the real thermodynamic losses cannot be quantified and the performance of the system components cannot be compared with each other. The exergy analysis answers these open questions.

In case TESP_y is used for the simulation of the thermodynamic performance, an exergy-based evaluation can be carried out using the class ExergyAnalysis. The students have to set up the exergy flow rates crossing the system boundary and define the exergetic fuel, the exergetic product and the exergetic loss of the overall process.

The buses for the net work rate and the fuel input are already defined. In case of fuel the chemical exergy must be considered. Additionally, an exergy loss bus must be defined for the exhaust gas.

The generic exergy analysis of TESP_y can be set up within a few lines of code, defining the above mentioned exergy flow rates and the ambient conditions.

```

from tespy.tools import ExergyAnalysis
ean = ExergyAnalysis(gas_turbine, E_P=[work_net], E_F=[ex_fuel], E_L=[ex_loss])
ean.analyse(pamb=1.013, Tamb=25)

```

The graphical evaluation of the exergy analysis can be done, for example, with a waterfall diagram. Penkuhn et al. [26] show this for a cogeneration plant. Another form of visualization is the representation of exergy rates in a Sankey diagram, often called a Grassmann diagram.¹⁴ The exergy analysis results are linked to the representation of the process flow diagram. The different material and energy streams of the flowsheet are brought to a uniform basis with the help of the exergy. The width represents the exergy rate. In addition, the exergy destruction is directly assigned to each component. Figure 10 shows this for the gas turbine process under investigation. The highest exergy destruction rate occurs in the combustion chamber, followed by the expander and air compressor. The exhaust gases represent an exergy loss for the overall process.

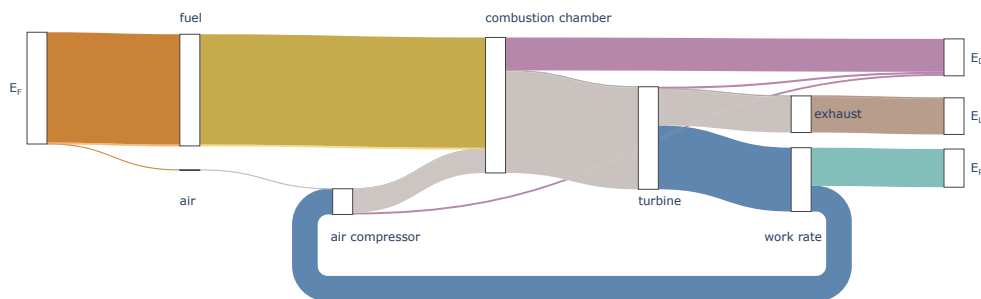


Figure 10: Grassmann diagram of gas turbine process showing the results of the exergy-based assessment

¹⁴Grassmann [27] transferred the idea of Sankey to the exergy concept. Exergy flow diagrams are therefore associated with his name.

The light tint indicates physical exergy, and the dark tint indicates chemical exergy, for example, visible in the fuel exergy flow rate or the exergy flow rate between the combustion chamber and the expander. It can be observed that all flows are either nearly pure chemical or pure physical exergy.

Lessons Learned from this task:

- Learn that TESP_y can do a lot of work for you.
- Relevant thermodynamic process parameters identified and numerically evaluated.
- Applied exergy analysis for a simple process for the first time, and identified the locations and magnitudes of real thermodynamic inefficiencies.

4.3. Highlight 3: Process Design Study for a Combined Heat and Power Plant

This example contains a process design study. A proposal for a power plant to supply electricity and district heating is to be developed. The heat source is a geothermal field.

The design proposal is to be modeled, simulated, and evaluated independently by the students. The design based on the principle of cogeneration is to be compared with a process design with complete condensation operation (without heat extraction). For this purpose, the energetic and exergetic efficiency of the overall process will be defined.

$$\eta = \frac{\dot{W}_{\text{net}} + \dot{Q}_{\text{DH}}}{\dot{H}_1 - \dot{H}_2} \quad (1)$$

$$\varepsilon = \frac{\dot{W}_{\text{net}} + (\dot{E}_{\text{FL}} - \dot{E}_{\text{RL}})}{\dot{E}_1 - \dot{E}_2} \quad (2)$$

All predefined parameters and boundary conditions can be found in Figure 11 and Table 3.

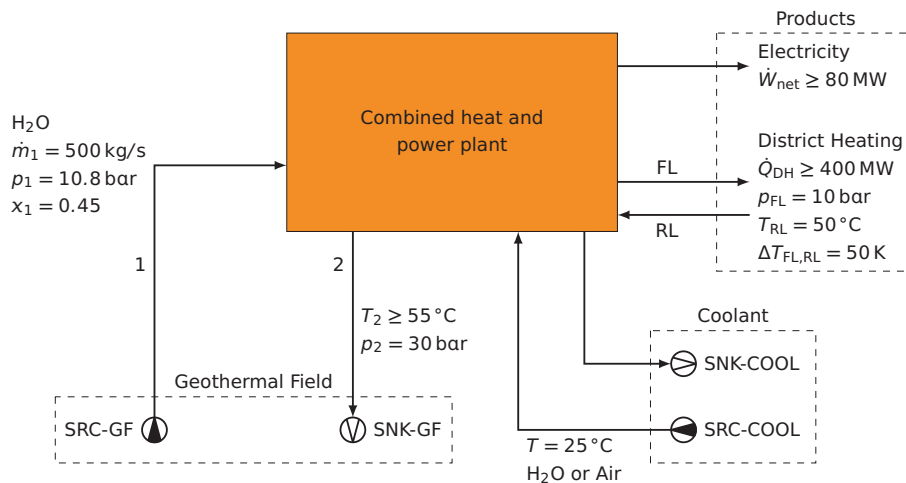


Figure 11: Process design study for a combined heat and power plant to provide electricity and district heating.

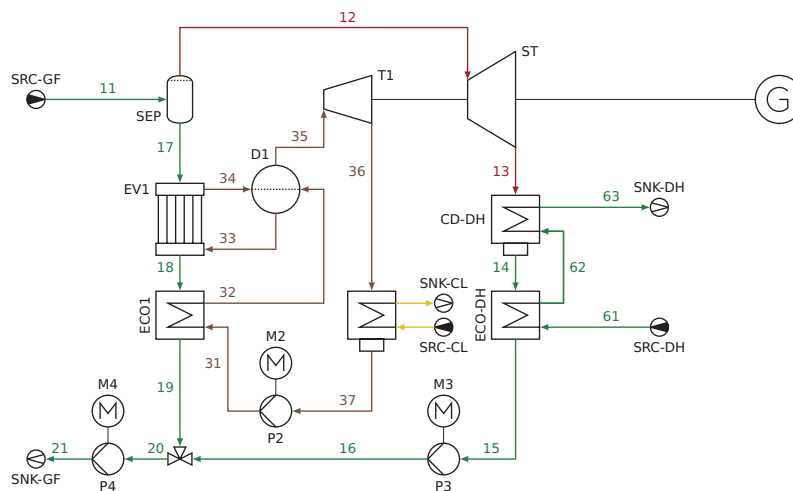
The following subtasks are to be solved within the scope of the assigned problem: Selection of suitable working fluids and selection and reasoning of the use of thermodynamic properties models for the simulation; creation of a process design, modeling, and simulation in TESP_y; definition and use of relevant variables for the evaluation of the process designs and subsequent comparison and discussion.

There is no generally valid standard solution for solving this task. Two possible designs with and without heat extraction are shown in Figure 12. In both variants, the heat flow of the liquid part of the geothermal fluid is absorbed by an upstream organic Rankine cycle (ORC) and used to produce electricity. The steam phase is expanded in a steam turbine. In the case of simultaneous provision of district heating, a heating condenser with preheating follows before the geothermal fluid is returned. In the design for electricity generation only, a second ORC is inserted after the steam turbine.

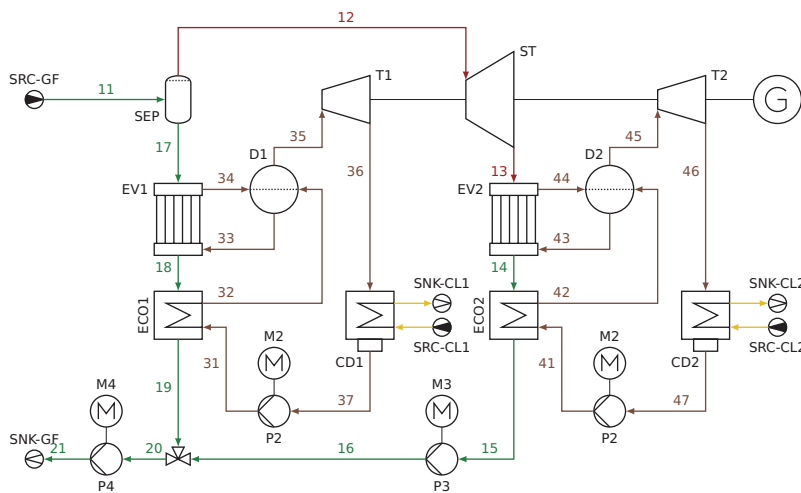
Both process designs are simulated and compared both energetically and exergetically. The results are shown in Table 4. According to Eq. (1), the energetic evaluation treats two different energy flows (electricity and heat)

Table 3: Process design study specifications and constraints

Component	Parameter	Unit	Value	
Auxiliary installations	Electrical power requirements	MW	3	
Generators	Mechanical-electrical efficiency	–	0.985	
Motors	Electrical-mechanical efficiency	–	0.975	
Steam turbines	Isentropic efficiency	–	0.9	
	Mechanical efficiency	–	0.99	
	Minimum steam quality	–	0.9	
Pumps	Isentropic efficiency	–	0.8	
	Mechanical efficiency	–	0.99	
Heat exchangers	Minimal temperature difference	liquid–liquid	K	10
		liquid–condensing	K	5
		liquid–steam	K	20
		liquid–gaseous	K	25



(a) Geothermal combined heat and power plant



(b) Geothermal power plant

Figure 12: Flowcharts of possible process designs

equally. The efficiency is significantly higher in the case of heat extraction. It gives the impression that the process is much closer to the thermodynamic optimum, which is not the case, as the comparison of the exergetic efficiencies shows. Here, both exergetic products can be directly converted into each other. The result is that the process designs have approximately similar range of exergetic efficiencies, and therefore the misestimation of the energetic evaluation can be revealed.

Table 4: Comparison of energetic and exergetic efficiency for developed process designs

Process design	Energetic efficiency η [%]	Exergetic efficiency ε [%]
Geothermal combined heat and power plant	80.4	70.3
Geothermal power plant	19.0	62.8

Lessons Learned from this task:

- The exergy analysis assists in the selection and improvement of a suitable process design.
- Different material and energy streams can be consistently evaluated based on exergy.
- Various process designs with different fuels or products remain comparable with the help of exergy.

5. Conclusion

The idea presented in this article is based on the notion that research and teaching should be accessible to all with as few barriers as possible. At the same time, the paper addresses the exergy concept, one of the essential analysis tools of thermodynamic engineering. Object-oriented programming with Python, particularly steady-state power plant simulation with open-source software, is linked to exergy as an established thermodynamic quantity in the concept presented.

A modular structure is developed to facilitate the integration of the resources into existing courses and for self-learners to choose their appropriate entry point. The modular structure covers exergy analysis of single component operation as the entry-level, where the fundamental understanding of the exergy concept is conveyed. Based on that, students will learn to model simple thermodynamic cycles and to carry out first and second-law analyses by evaluating the simulation results. It is further supported by applying parametric analysis to understand the connection between individual component parameters to the thermodynamic performance of the overall process.

Further processing possibilities arise from the potential integration of additional exercises, the extension to the exergoeconomic analysis [28], the exergoenvironmental analysis [29], or the advanced exergy analysis [30]. At the same time, all interested students, researchers, and practitioners are invited to try the course, and to contribute by making comments or reporting bugs and developing it further collaboratively.

CRedit author statement

Mathias Hofmann: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Francesco Witte:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data Curation, Writing - Review & Editing **Malte Fritz:** Software, Validation, Formal analysis, Data Curation **Jonas Freißmann:** Software, Validation, Formal analysis, Data Curation **Ilja Tuschy:** Review & Editing, Supervision **George Tsatsaronis:** Review & Editing, Supervision

Malte Fritz and **Jonas Freißmann** contributed equally to this work.

Acknowledgments

The authors would like to thank Björn Kiehne (Berliner Zentrum für Hochschullehre) for his remarks on Section 1. and 2. Mathias Penkuhn (Technische Universität Berlin) has to be credited for the conceptual design of the task given in Section 4.3.

Nomenclature

Abbreviations

- A Air
- AC Air compressor
- CC Combustion chamber

CD Condenser
CL Coolant
D Drum
DH District heating
E Exhaust
ECO Economizer
EXP Expander
EV Evaporator
F Fuel
FL Feed line
G Generator
GF Geothermal fluid
GT Gas turbine
M Motor
ORC Organic Rankine cycle
P Pump
SEP Separator
SNK Sink
SRC Source
ST Steam turbine
T Turbine
RL Return line

Letter symbols

\dot{E} Exergy rate, W
 \dot{H} Enthalpy rate, W
 \dot{m} Mass flow rate, kg/s
 p Pressure, bar
 \dot{Q} Heat rate, W
 r Ratio, –
 T Temperature, °C
 w Specific work, kJ/kg
 \dot{W} Work rate, W
 x Quality, –
 y_i Mass fraction, –

Greek symbols

Δ Difference
 ε Exergetic efficiency
 η Energetic efficiency

Subscripts and superscripts

0 At ambient state
 D Destruction
 F Fuel
 i Stream
 p Pressure
 P Product
 s Isentropic

References

- [1] D. Abdrasheva, M. Escribens, E. Sabzalieva, D. V. do Nascimento, and C. Yerovi. *Resuming or Reforming? Tracking the global impact of the COVID-19 pandemic on higher education after two years of disruption*. Retrieved: January 27th, 2023. 2022. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000381749>.
- [2] K. H. Mok. "Impact of COVID-19 on Higher Education: Critical Reflections". In: *High. Educ. Policy* 35 (2022), pp. 563–567. DOI: 10.1057/s41307-022-00285-x.
- [3] F. S. von Thun. "Wie gestalte ich meine Vorlesung – und halte die Hörschaft und mich selbst bei Laune?" In: *Die Qualität Akademischer Lehre*. Ed. by M. Merkt and K. Mayrberger. in German. Innsbruck, Austria: Studienverlag, 2007, pp. 115–131.
- [4] L. W. Anderson and D. R. Krathwohl, eds. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman, 2001.
- [5] Z. Rant. "Exergie, ein neues Wort für technische Arbeitsfähigkeit". In: *Forsch. Ing.-Wes.* 22.1 (1956). in German, pp. 36–37.
- [6] T. J. Kotas. *The Exergy Method of Thermal Plant Analysis*. London, UK: Butterworths, 1985.
- [7] W. Fratzscher, M. Brodjanskij, and K. Michalek. *Exergie*. in German. Leipzig, Germany: Verlag für Grundstoffindustrie, 1986.
- [8] A. Bejan, G. Tsatsaronis, and M. Moran. *Thermal design and optimization*. New York, USA: Wiley, 1996.
- [9] J. Szargut. *Exergy Method*. Southampton, UK: WIT Press, 2005.
- [10] W. Rüegg, ed. *Geschichte der Universität in Europa*. Vol. 1, Mittelalter. in German. München, Germany: Beck, 1993.
- [11] R. A. Müller. *Geschichte der Universität*. in German. München, Germany: Callwey, 1990.
- [12] M. J. Moran, H. N. Shapiro, D. D. Boettner, and M. B. Bailey. *Fundamentals of Engineering Thermodynamics*. 9th ed. New York, USA: John Wiley, 2018.
- [13] H. D. Baehr and S. Kabelac. *Thermodynamik*. in German. Berlin, Germany: Springer, 2016. DOI: 10.1007/978-3-662-49568-1.
- [14] E. Matthes. *Python Crash Course*. San Francisco, USA: No Starch Press, 2023.
- [15] A. Galea. *Beginning Data Analysis with Python and Jupyter*. Birmingham, UK: Packt Publishing, 2018.
- [16] F. Witte and I. Tuschy. "TESPy: Thermal Engineering Systems in Python". In: *J. Open Source Softw.* 5 (2020), p. 2178. DOI: 10.21105/joss.02178.
- [17] F. Witte, M. Hofmann, J. Meier, I. Tuschy, and G. Tsatsaronis. "Generic and Open-Source Exergy Analysis—Extending the Simulation Framework TESPy". In: *Energies* 15 (2022), p. 4087. DOI: 10.3390/en15114087.
- [18] M. Hofmann, F. Witte, K. Shawky, I. Tuschy, and G. Tsatsaronis. "Thermal Engineering Systems in Python (TESPy): The implementation and validation of the chemical exergy". In: *Proceedings of ECOS 2022*. Copenhagen, Denmark, 2022, pp. 257–269.
- [19] S. Pfenninger, J. DeCarolis, L. Hirth, S. Quoilin, and I. Staffell. "The importance of open data and software: Is energy research lagging behind?" In: *Energy Policy* 101 (2017), pp. 211–215. DOI: 10.1016/j.enpol.2016.11.046.
- [20] *The Journal of Open Source Education*. Retrieved: March 23rd, 2023. URL: <https://jose.theoj.org/>.
- [21] R. Morrison. *einfach : an energy system model for education*. 2020. DOI: 10.5281/zenodo.7746137.
- [22] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort. "Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp". In: *Ind. Eng. Chem. Res.* 53.6 (2014), pp. 2498–2508. DOI: 10.1021/ie4033999.
- [23] F. Biscani and D. Izzo. "A parallel global multiobjective framework for optimization: pagmo". In: *J. Open Source Softw.* 5.53 (2020), p. 2338. DOI: 10.21105/joss.02338.
- [24] J. Ahrendts. "Reference states". In: *Energy* 5 (1980), pp. 666–677. DOI: 10.1016/0360-5442(80)90087-0.
- [25] T. Morosuk and G. Tsatsaronis. "Splitting physical exergy: Theory and application". In: *Energy* 167 (2019), pp. 698–707. DOI: 10.1016/j.energy.2018.10.090.
- [26] M. Penkuhn, M. Hofmann, S. Meinke, and C. Lösche. "Prozessbewertung und Exergieanalyse für ein Heizkraftwerk". In: *Kraftwerkstechnisches Kolloquium*. Dresden, Germany, 2021, pp. 1–12. DOI: 10.14279/depositonice-15966.
- [27] P. Grassmann. "Die Exergie und das Flussbild der technisch nutzbaren Leistung". In: *Allg. Wärmetechn.* 9.4/5 (1959). in German, pp. 79–86.
- [28] G. Tsatsaronis. "Thermoeconomic analysis and optimization of energy systems". In: *Prog. Energy Combust. Sci.* 19 (1993), pp. 227–257. DOI: 10.1016/0360-1285(93)90016-8.
- [29] L. Meyer, G. Tsatsaronis, J. Buchgeister, and L. Schebek. "Exergoenvironmental analysis for evaluation of the environmental impact of energy conversion systems". In: *Energy* 34 (2009), pp. 75–89. DOI: 10.1016/j.energy.2008.07.018.
- [30] M. Penkuhn and G. Tsatsaronis. "A decomposition method for the evaluation of component interactions in energy conversion systems for application to advanced exergy-based analyses". In: *Energy* 133 (2017), pp. 388–403. DOI: 10.1016/j.energy.2017.03.144.

A Appendix: Open-Cycle Gas Turbine – TESPpy Source Code

```
from tespy.networks import Network
from tespy.components import Sink, Source, Compressor, DiabaticCombustionChamber, Turbine
from tespy.connections import Connection, Bus

# general configurations
# substances, fluids, network, units
fluid_list = ['N2', 'O2', 'Ar', 'CO2', 'H2O', 'CH4']
gas_turbine = Network(fluids=fluid_list, p_unit='bar', T_unit='C', h_unit='kJ / kg')

# composition of given fluids
air = {'N2': 0.7551, 'O2': 0.2314, 'Ar': 0.0129, 'CO2': 0.0006, 'H2O': 0, 'CH4': 0}
fuel = {'N2': 0, 'O2': 0, 'Ar': 0, 'CO2': 0, 'H2O': 0, 'CH4': 1}

# define sources and sinks
src_air = Source('air')
src_fuel = Source('fuel')
snk_exhaust = Sink('exhaust')

# define components
cmp_AC = Compressor('air compressor')
cmp_CC = DiabaticCombustionChamber('combustion chamber')
cmp_EX = Turbine('expander')

# define connections
c1 = Connection(src_air, 'out1', cmp_AC, 'in1', label='1')
c2 = Connection(cmp_AC, 'out1', cmp_CC, 'in1', label='2')
c3 = Connection(src_fuel, 'out1', cmp_CC, 'in2', label='3')
c4 = Connection(cmp_CC, 'out1', cmp_EX, 'in1', label='4')
c5 = Connection(cmp_EX, 'out1', snk_exhaust, 'in1', label='5')

# add connections to network
gas_turbine.add_conns(c1, c2, c3, c4, c5)

# parameter of components
cmp_AC.set_attr(eta_s=0.85, pr=18)
cmp_CC.set_attr(eta=0.98, pr=0.95)
cmp_EX.set_attr(eta_s=0.9)

# parameter of connections
c1.set_attr(p=1.013, T=25, fluid=air)
c3.set_attr(p=20, T=25, fluid=fuel)
c4.set_attr(T=1400)
c5.set_attr(p=1.013)

# busses
work_net = Bus('work netto')
fuel_in = Bus('fuel input')

work_net.add_comps(
    {'comp': cmp_AC, 'base': 'bus', 'char': 1},
    {'comp': cmp_EX, 'char': 1})

fuel_in.add_comps(
    {'comp': cmp_CC, 'base': 'bus'})

gas_turbine.add_busses(work_net, fuel_in)

# parameter of busses
work_net.set_attr(P=-150e6)

# solve network
gas_turbine.solve('design')

# print network results
gas_turbine.print_results()
```