

ReSSSPI: The network of simulated solar systems for industrial processes

Miguel Frasset¹, Joey Bannenberg², Yago N. Vila³ and Manuel Silva⁴

¹ Solatom CSP S.L, Valencia (Spain)

² Fontys University of applied sciences, Eindhoven (Netherlands)

³ ETH, Zürich (Switzerland)

⁴ University of Sevilla, Sevilla (Spain)

Abstract

During 2017, the Spanish Ministry of Economy, Industry and Competitiveness through the technology platform “Solar Concentra”, launched a field study to identify the one hundred Spanish industries with the highest potential of using concentrating solar collectors to meet their thermal energy demand. In order to evaluate both the energy and economic performance in each case, an open-source solar simulator for industrial processes called ReSSSPI has been developed during this study. ReSSSPI is an online calculator (OC) written in Python 3 that estimates the economic and energy performance of solar process heat applications. ReSSSPI’s performance model is based on SHIPcal code, one of the earliest OCs for solar heat applications in industrial processes. In one year, ReSSSPI has performed more than 267 simulations from Spanish industry users, creating a network of simulated solar plants across the Spanish territory.

Keywords: RESSSPI; solar; simulator; industrial; process; SHIPcal; online; calculator

1. Introduction

During 2017, the Spanish Ministry of Economy, Industry and Competitiveness through the Spanish technology platform for Solar Concentration “Solar Concentra”, launched a field study to assess the use of concentrated solar technologies for heat generation in industrial processes (Frasset, 2017). The goal of this study aimed to identify the Spanish industries with the highest potential to use concentrated solar collectors to meet their thermal energy demand. To assess the potential of every industry, a detailed set of information was obtained from industries in 8 different sectors. A total of two hundred industries were studied. In most cases, the information needed was gathered through telephone interviews and field visits.

In order to establish a common framework with which benchmark every case, an open-source solar simulator called ReSSSPI was developed during the study. ReSSSPI is an online calculator (OC) written in Python 3 capable of estimating both the energy performance, and the financial aspects of a large range of solar process heat applications.

2. Online calculators as benchmarking tool

When simulating solar thermal applications for industrial processes, there are several tools available in the market. These tools range from very simple Excel sheets, to quite complex but detailed software such as TRNSYS or PolySun. Due to time constraints, and the diversity of integration concepts found during the project, the ideal solution was a compromise between the flexibility of commercial software like the ones mentioned before, in which users are not limited to use pre-designed models, and the speed and simplicity of Excel sheets. In addition, since the goal of the project is to identify the industries with the highest potential, rather than to perform very

accurate simulations of every one of them, certain degree of uncertainty in the results is considered acceptable. Upon these considerations, the use of an OC was the final decision.

The use of OCs has a long tradition in more mature sectors like photovoltaic industry (PV). Nowadays hundreds of PV calculators can be found online (e.g NREL's PVWatts (Dobos, 2013) or SolarCalculator website <https://solarcalculator.com.au>). Although it is possible to find some OCs for solar heat for industrial processes (SHIP), such as SHIP Design Tool or Appsol. All of them proved to be inadequate for this study due to the limited integrations concepts available, and the lack of solar radiation sources regarding the Spanish territory.

SHIPcal code was released in 2015 (Frasquet, 2016) as an open source OC for solar heat applications in industrial processes. This tool came with an extensive data base of meteorological files for the Spanish territory, and some of the integration concepts proposed by the IEA SHC Task 49 (Muster et al., 2015). The results of SHIPcal were validated using TRNSYS, showing discrepancies in the energy production up to 10% during cloudy days (cumulative values), and 3% during sunny conditions. Although this level of accuracy made SHIPcal unsuitable to be used as a design program, is enough for the purpose of this project. Unfortunately, since the development of SHIPcal ceased one year after its release due to financial constraints, only the backend code (not the fully working tool) was available. Therefore a new OC called ReSSSPI, was developed during the project, using a completely new front-end and most of the SHIPcal's backend.

3. ReSSSPI online calculator

ReSSSPI has been the OC used as benchmarking tool during the project. Due to the size and heterogeneity of the group of industries studied, ReSSSPI has been designed to provide simple yet flexible simulations, of cost and energy performance, in a reasonable time span. To facilitate its use, during and after the project, ReSSSPI runs over a user-friendly website available online at www.ressspi.com.

3.1. Improvements over SHIPcal original approach

In addition to be tailored to fulfill the requirements of the project, ReSSSPI has also solved some of the limitations of the original SHIPcal approach.

SHIPcal was conceived as open-source initiative, in order to boost the collaboration of all stakeholders in the SHIP business. For these matters, the code was fully written in GNU Octave. Since Octave is a free and open-source programming language compatible with Matlab, the initial idea was to allow an easy transition from licensed software such Matlab (very popular in the scientific community), to free and compatible alternatives. However, this decision proved to be counterproductive due to the low popularity of Octave. In fact, based on the PYLP index 2018 (<http://pypl.github.io/PYPL.html>), an index that ranks the popularity of programming languages based on Google Trends, Matlab ranks in tenth position, while Octave is not even in the top 50. In ReSSSPI, all the code has been translated from Octave into Python, which not only ranks the first one in the PYLP index, it is free and open-source like Octave.

Another decision that significantly limited the development of SHIPcal, was the version control system used for tracking changes in the code. SHIPcal relied on a simple mailing list in which all the researchers contributing to the development of the code, posted their changes. This quickly spiraled out of control generating inefficiencies and overlaps. ReSSSPI however uses Git as version control system. Git was created by Linus Torvalds in 2005 for managing the collaborative development of the open-source software Linux, and since then, it has successfully managed the changes in the source code, applied by more than 15600 external contributors (Corbet et al., 2010). The source code of ReSSSPI is freely available at GitHub (<https://github.com/mfrasquet/>), a public repository of open-source projects using Git. From there, any interested user can clone the source code and start collaborating in its development.

3.2. ReSSSPI input form

As in most OCs, ReSSSPI uses a simple online form to gather the data needed for the simulation. This input form has 5 primary variables, and 19 secondary ones. Although all the variables are needed to perform a simulation, ReSSSPI can suggest default values for the secondary variables, as long as the primary variables are provided by the user. The primary variables are: Industry sector, size of the industry, type of fossil fuel currently used in the

industry, annual thermal energy demand, and location. The same way SHIPcal suggested values from the “Matrix of Industrial Process Indicators” of the Greenfood platform (<http://www.green-foods.eu/>) when some variables were left blank, once the primary variables are provided in ReSSSPI, it suggests default values for the missing secondary variables. But instead of using external sources like in SHIPcal, it generates default values using similar projects in its database.

ReSSSPI comes with meteorological and solar radiation data for 50 cities in Spain and several international locations. When data is not available for a specific location, it can be requested using the online website. Although the main source of the meteo and solar database is Meteonorm 7.2, in particular cases (like in the case of some Spanish cities) another sources have also been used. More information regarding meteo source can be found in ReSSSPI manual (Frasquet, 2018).

Once the input form is filled, and before the simulation starts, ReSSSPI checks the input set using a precautionary firewall. This firewall checks if the variables introduced are technically and/or thermodynamically feasible. If the user provides an outlet temperature lower than the input temperature, or the fluid chose is not compatible with the integration concept, for example, ReSSSPI stops the process avoiding a potential convergence problem when simulating.

3.3. Design process and simulation

ReSSSPI implements two design methods: Manual and Automatic. In the manual mode, the user introduces the design parameters manually (i.e. Number of loops, number of collectors per loop, integration concept and storage capacity if needed), whereas in the automatic mode, ReSSSPI finds the optimal configuration by itself. At the time of writing, ReSSSPI offers the possibility of choosing between 7 different industrial integration concepts proposed by Muster et al. (2015). It also enables the use of thermal storage depending on the integration concept selected, and some other specific integration concepts, found recursively during the project. Table 1 show the characteristics and the current development of the integration concepts available in ReSSSPI.

Tab. 1: Current development of integration concepts available in ReSSSPI

Solar heat integration concept (* not included in Muster et al. (2015))	Level of integration	Heat transfer medium	Integration description	Storage	Current status in ReSSSPI
SL_L_P	Supply	Water/Oil	Parallel integration (Direct)	No	Operative
SL_L_PS	Supply	Water/Oil	Parallel integration (Direct)	Yes	Operative
SL_L_RF	Supply	Water/Oil	Return flow boost	No	Operative
SL_L_S	Supply	Water/Oil	Heating of storages	Yes	Operative
SL_S_PD	Supply	Steam	Parallel integration (Direct)	No	Operative
SL_S_PI	Supply	Steam	Parallel integration (Indirect)	No	Being tested
SL_S_FW	Supply	Steam	Heating of feedwater	No	Operative
SL_S_FWS	Supply	Steam	Heating of feedwater with intermediate storage	Yes	Operative
SL_S_MW	Supply	Steam	Heating of make-up water	No	Operative
PM_E_PM	Process	Water/Oil	Heating of process medium	No	Operative
PM_E_PM_Past*	Process	Water	Integration for pasteurizators with intermediate storage	Yes	Being tested
PM_E_PM_Sew*	Process	Water/Oil	Integration for Sewage plants with intermediate storage	Yes	Operative

During the project the manual mode was used in the vast majority of the cases, since the users of the code were solar specialists. However, since ReSSSPI is meant to be used by public with all types of technical background, an automatic design method was needed.

When the automatic method is chosen, ReSSSPI creates a list of potential configurations depending on the input received. For example, if the process introduced in the form requires the use of steam, it creates potential configurations using all the possible alternatives: SL_S_PD, “Direct solar steam generation” and SL_S_FW, “solar heating of boiler feed water”. If the process uses hot water/thermal oil as working fluid, it creates the list of candidates combining the following integration concepts: SL_L_RF, “solar return flow boost”, SL_L_P, “parallel integration” and PL_E_PM, “external HEX for heating of product or process medium”. Once the list of potential configurations is generated, it performs a generic annual simulation for every configuration and selects the one which delivers the minimum Levelized Cost of Thermal Energy (LCOE_{th}) (IRENA, 2012). The LCOE_{th} is calculated using eq. 1.

$$LCOE = \frac{\sum_{t=1}^n \frac{I_t + M_t + F_t}{(1+r)^t}}{\sum_{t=1}^n \frac{E_t}{(1+r)^t}} \quad (\text{eq. 1})$$

The configuration with the lowest LCOE_{th} is then simulated on an hourly-basis, combining different solar field designs (number of collectors in series per loop, and number of loops in the plant), with different sizes of thermal storage (if the selected configuration allows thermal storage).

Only one annual simulation is performed when the manual simulation procedure is used, thus it gives short simulation times (aprox. 30 seconds). In automatic simulations however, ReSSSPI needs to simulate every configuration listed as a potential candidate. And once the configuration is proved to deliver the lowest LCOE_{th}, ReSSSPI still needs to optimize the solar field and the size of the storage (if available). Currently, this process takes almost 23min and limits its potential application as real time online calculator. During the first half of 2018, a collaboration with Fontys University of Applied Sciences studied the use of different optimization algorithms to speed up the design optimization process. The following optimization algorithms were tested: Greedy search algorithms such as local search (Michiels, et al. 2007) and hill-climbing (Romeo, 1984), non-greedy search algorithms that use cooling schedules such as simulated annealing (Du and Swamy, 2016) and threshold accepting (Dueck and Scheuer, 1990), non-greedy search algorithm that use memorizing strategies such as Tabu search (Gendreau, 1995) and late acceptance hill-climbing (Burke and Bykov, 2012), and combinations of these algorithms. This research led to several conclusions, where the most important conclusions are that the non-greedy algorithm local search is preferred when searching the minimum LCOE_{th} in search spaces where it is known that only one minimum exists, while the tabu search and the hill climbing algorithms are preferred to find the global minimum for search spaces that contain several local minima for small and large search spaces respectively.

3.4. Generic vs third party libraries

Although ReSSSPI source code comes with generic libraries for the solar collector optics and economics, it is prepared to include third-party libraries. When using the generic library, the collector data needed in the simulation (Aperture area, length of collector, IAM, etc.) comes from publicly available literature (Schenk, 2014). In a similar way, the required economic data in the generic libraries has been gathered from different sources (IDEA, 2010). When using third-party libraries however, both solar optics and economic data are provided by companies with commercial collectors in operation.

The solar energy yield is calculated following the guidelines developed during Guismo project (Eck, et al., 2013), in which the energy output of the solar field is obtained using eq.2 and eq.3:

$$\dot{Q}_{solar} = A_{net} \cdot DNI \cdot \eta_{opt,0} \cdot IAM(\theta) \cdot \eta_{clean} \quad (\text{eq. 2})$$

$$\eta_{opt,\theta} = \rho \cdot \tau \cdot \alpha \cdot \gamma_{IC} \quad (\text{eq. 3})$$

Where *IAM* is the incidence angle modifier which describes the collector losses depending on the incidence angle

θ (eq.4). Note that in the case of linear Fresnel two IAM must be evaluated to take into consideration both transversal and longitudinal incidence angle.

$$IAM(\theta) = \frac{\eta_{opt}(\theta)}{\eta_{opt,0}(\theta=0)} \quad (\text{eq. 4})$$

In this project, a third-party library provided by Solatom has been used to perform all simulations. Solatom’s collector uses linear Fresnel optics, and comes in modules of 6m long, see Figure 1. This third-party library is also the default library in the front-end version available on-line.

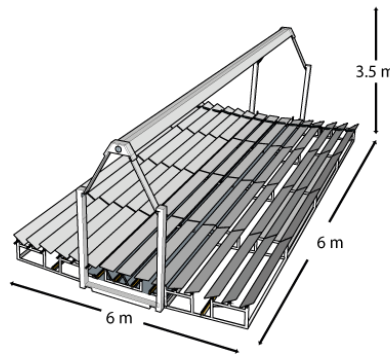


Fig. 1: Geometric dimensions of Solatom’s linear Fresnel collector used in ReSSSPI

3.5. Finance model

ReSSSPI currently includes two finance models: turnkey and ESCO. In both cases, the model provides the total investment needed, annual savings, free cash flows and return of investment. ReSSSPI considers a set of hypotheses needed for the financial calculation such as interest rate, mid-term evolution of fossil fuel price, price of CO₂ ton, etc. These variables can be modified or leave fixed on their default values (see Frasquet, (2018))

3.6. Output of results

Once the simulation is completed, ReSSSPI outputs the results obtained through a plain text summary and a set of images grouped in 3 categories: Solar plant, energy production and economics. In the solar plant category, ReSSSPI shows a scheme of the integration concept selected, and the characteristics of the process fluid. When water or steam is used, ReSSSPI outputs T-s and h-s diagrams with the fluid evolution. When thermal oil is used, only the oil’s property graphs are provided. See Figure 2

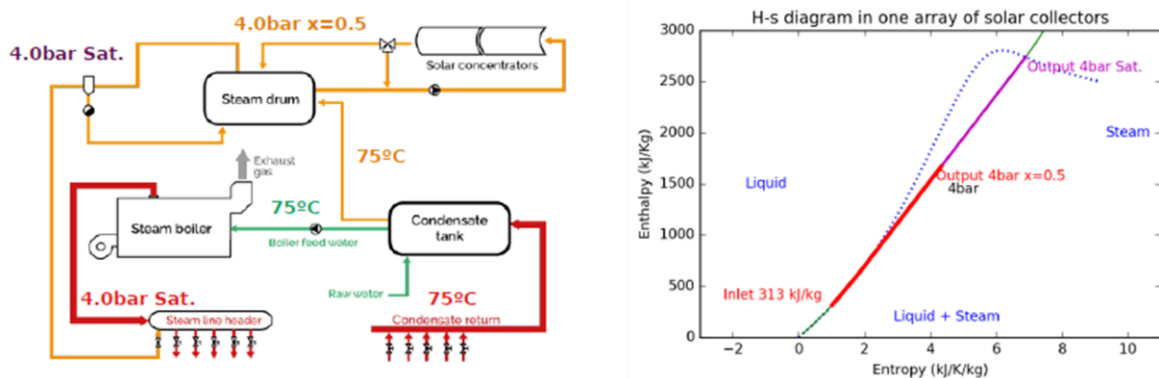


Fig. 2: (Left) Example of diagram ReSSSPI provides when the SL_S_PD configuration concept is used in the simulation (Right) Example of one h-s process diagram provide as output

Initially ReSSSPI only plotted monthly energy production. But during the project, after meeting with many industry managers, it was clear their difficulties understanding the role of thermal storage in the output yield of the solar system. For that reason, the evolution of the state of charge during the first week of summer/winter, has been included. As Figure 3 shows, the plot also provides information regarding the amount of energy going into the storage (Charging) and going out of the system (Discharging).

Regarding the financial aspects, ReSSSPI provides a visual image of how the sum of free cash flows generated by the energy savings breaks even when it equals the initial investment.

As in the case described before, where solar collectors could be determined using generic or third-party libraries, ReSSSPI is also capable of using modules written by third-parties when printing the results. In fact, the online website has the possibility of using a module written by Solatom capable of generating pdf templates with all the images and results that ReSSSPI provides.

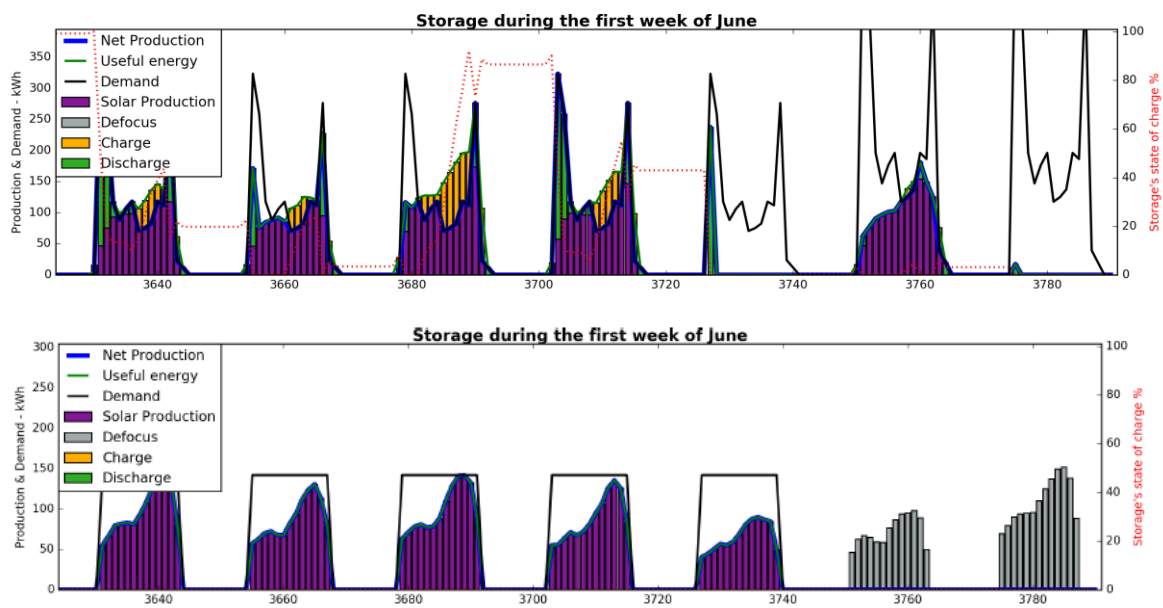


Fig. 3: State of the Charge of the thermal storage and energy inputs/outputs in the system. (Top) Solar plant with storage and demand during the whole week. (Bottom) Solar plant without storage and demand only the first 5 days of the week (Industry has no process during the weekends)

4. Results and validation

ReSSSPI code was developed during 2017 as the benchmarking tool needed to assess the potential of the industries studied during the project. From there, it has performed more than 267 simulations creating a significant data base of simulated solar projects. Figure 4 shows the location of the simulations performed since the origin of the tool. In June 2018, ReSSSPI included the automatic design method to allow users with no solar background to perform simulations through its website. Currently, every interested stakeholder can perform an annual simulation and benchmark the results against its database.

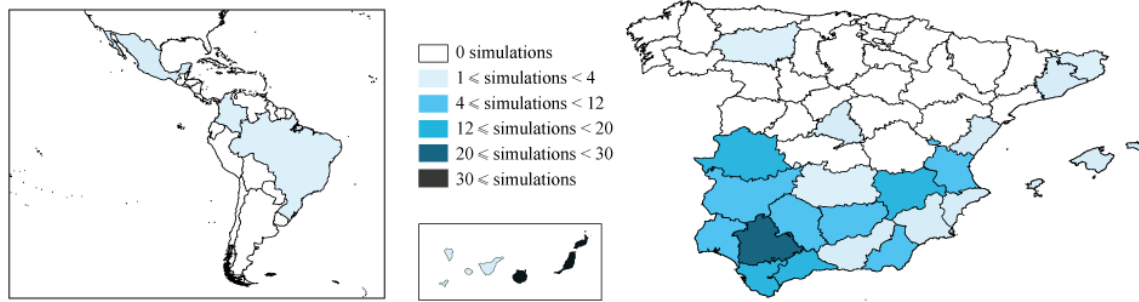


Fig. 4: State of the Charge of the thermal storage and energy inputs/outputs in the system

Although ReSSSPI still relays in SHIPcal original code, and this one has only been validated using theoretical simulations in TRNSYS, the validation with real operation data is already taking place on a Spanish agrochemical factory called Dadelos. Dadelos (Figure 5) is using a small linear Fresnel solar field (72m²) to provide pressurized hot water to its chemical reactors. The test campaign will be completed during the first quarter of 2019, so data during both summer and winter will be obtained.

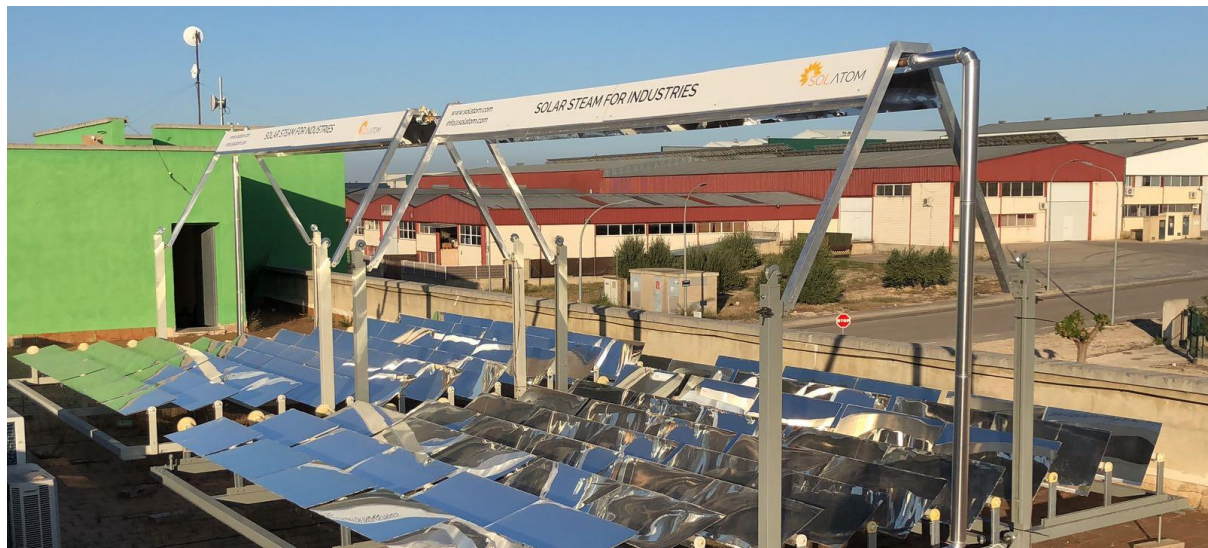


Fig. 5: Linear Fresnel solar field where the validation of ReSSSPI is being carried out

5. Conclusions

Online calculators have proven to be a useful tool for performing quick calculations of solar energy projects. Due to the characteristics of the study lead by Solar Concentra, in which two hundred Spanish industries were analyzed, current OCs were not flexible enough, and a new code called ReSSSPI, was developed. Initially based on SHIPcal's engine, ReSSSPI has solved some of SHIPcal's original limitations, and added more integration concepts to gain flexibility. Since its release in 2017, it has performed more than 267 simulations, creating a significant data base of simulated solar projects. Both the source code and the front-end are available online and open to the general public, which now can perform annual simulations and benchmark the results against its database.

6. Acknowledgements

The author would like to thank the support of the Spanish technology platform for concentrating solar power (Solar Concentra) and all the researchers and students involved in the development of ReSSSPI.

7. References

- Burke E.k., Bykov Y., 2012. The Late Acceptance Hill-Climbing Heuristic.
- Corbet J., Kroah-Hartman G., McPherson A., 2010. Linux Kernel Development. The Linux Foundation. <https://www.static.linuxfound.org/docs/lf_linux_kernel_development_2010.pdf> (accessed 13.06.18)
- Dobos, A. 2013. PVWatts Version 1 Technical Reference. NREL Technical Report NREL/TP-6A20-60272
- Du K., Swamy M.N., 2016. Search and Optimization by Metaheuristics. Springer.
- Dueck, G., Scheuer T., 1990. Threshold Accepting. A General Purpose Optimization Algorithm Superior to Simulated Annealing. Journal of Computational Physics 90, 161-175.
- Eck M., Hirsch T., Feldhoff F., Kretschmann D., Dersch J., Gavilan Morales A., Gonzalez-Martinez L., Bachelier C., Platzer W., Riffelann K.-J., Wagner M., 2013. Guidelines for CSP yield analysis –optical losses of line focusing systems; definitions, sensitivity analysis and modeling approaches. SolarPaces 2013. Energy Procedia 49 (2014) 1318 – 1327
- Frasquet M., 2016 SHIPcal: Solar Heat for Industrial Processes Online Calculator, Energy Procedia, Volume 91, 2016, Pages 611-619, ISSN 1876-6102
- Frasquet M., 2017. Estudio geolocalizado del potencial de aplicaciones de calor solar de proceso en media temperatura. Solar Concentra. Madrid <<http://www.solarconcentra.org/estudio-geolocalizado-del-potencial-de-aplicaciones-de-calor-solar-de-proceso-en-media-temperatura/>> (accessed 01.06.18)
- Frasquet M., 2018. ReSSSPI reference manual. <<http://www.ressspi.com/static/documentation/manual.pdf>>
- Gendreau M., 1995. An Introduction to Tabu Search. Handbook of Metaheuristics. Springer, Boston, MA
- IDAE (Instituto para la Diversificación y Ahorro de Energía), 2010. Evaluación del potencial de la energía solar térmica en el sector industrial. <http://www.idae.es/uploads/documentos/documentos_11227_e8_ST_industria_bf2e9296.pdf> (accessed 25.06.18)
- IRENA (International Renewable Energy Agency), 2012. Renewable Energy Technologies: Cost Analysis Series. Concentrating Solar Power. <http://www.irena.org/documentdownloads/publications/re_technologies_cost_analysis-csp.pdf> (accessed 25.06.18)
- Michiels W., Aarts E., Korst J., 2007. Theoretical Aspects of Local Search. Springer Science & Business Media. Berlin
- Muster B., Ben Hassine I., Helmke A., Heß S., Krummenacher P., Schmitt B., Schnitzer H., 2015. Solar Process Heat for Production and Advanced Applications. Integration Guideline. IEA
- Romeo F. I., 1984. Probabilistic hill climbing algorithms: Properties and applications. Unknown
- Schenk H., Hirsch T., Feldhoff F., Wittman M., 2014. Energetic comparison of linear Fresnel and parabolic trough collector systems. Journal of Solar Energy Engineering. < http://www.idae.es/uploads/documentos/documentos_11227_e8_ST_industria_bf2e9296.pdf> (accessed 25.06.18)

8. Appendix: Unites and Symbols

Table 1: Units and symbols for LCOE_{th} calculation

Quantity	Symbol	Unit
Investment expenditures in year t	I_t	€
O&M expenditures in year t	M_t	€
Fuel expenditures in year t	F_t	€
Energy generation in year t	E_t	kWh
Discount rate	r	
Life of the system	n	years

Table 2: Units and symbols for energy yield calculation

Quantity	Symbol	Unit
Net area of mirrors	A_{net}	m ²
Direct Normal Irradiation	DNI	W m ⁻²
Peak Optic Efficiency	$\eta_{opt,0}$	
Receiver tube solar absorptivity	α	
Mirrors solar reflectivity	ρ	
Geometric intercept factor of the collector	γ_{IC}	
Glass envelope solar transmissivity	τ	
Incidence angle	θ	°
Optic Efficiency	η_{opt}	
Incidence Angle Modifier	IAM	
Efficiency due to soiling of mirrors.	η_{clean}	