

SolarTherm: A flexible Modelica-based simulator for CSP systems

Cite as: AIP Conference Proceedings **1850**, 160026 (2017); <https://doi.org/10.1063/1.4984560>
Published Online: 27 June 2017

Paul Scott, Alberto de la Calle Alonso, James T. Hinkley, and John Pye



View Online



Export Citation

ARTICLES YOU MAY BE INTERESTED IN

[Techno-economic analysis of concentrated solar power plants in terms of levelized cost of electricity](#)

AIP Conference Proceedings **1850**, 160018 (2017); <https://doi.org/10.1063/1.4984552>

[Optical performance considerations for analysis and simulation of power tower plants](#)

AIP Conference Proceedings **1850**, 160021 (2017); <https://doi.org/10.1063/1.4984555>

[Development of a CSP plant energy yield calculation tool applying predictive models to analyze plant performance sensitivities](#)

AIP Conference Proceedings **1850**, 160012 (2017); <https://doi.org/10.1063/1.4984546>

Lock-in Amplifiers
... and more, from DC to 600 MHz



SolarTherm: A Flexible Modelica-based Simulator for CSP Systems

Paul Scott¹, Alberto de la Calle Alonso², James T Hinkley² and John Pye^{1,a)}

¹*Solar Thermal Group, Research School of Engineering, Australian National University, Canberra, Australia.*

²*CSIRO Energy Technology, Newcastle, Australia.*

^{a)}Corresponding author: john.pye@anu.edu.au

Abstract. Annual performance simulations provide a valuable tool for analysing the viability and overall impact of different concentrating solar power (CSP) component and system designs. However, existing tools work best with conventional systems and are difficult or impossible to adapt when novel components, configurations and operating strategies are of interest. SolarTherm is a new open source simulation tool that fulfils this need for the solar community. It includes a simulation framework and a library of flexible CSP components and control strategies that can be adapted or replaced with new designs to meet the special needs of end users. This paper provides an introduction to SolarTherm and a comparison of models for an energy-based trough system and a physical tower system to those in the well-established and widely-used simulator SAM. Differences were found in some components where the inner workings of SAM are undocumented or not well understood, while the other parts show strong agreement. These results help to validate the fundamentals of SolarTherm and demonstrate that, while at an early stage of development, it is already a useful tool for performing annual simulations.

INTRODUCTION

Simulators for CSP annual performance, such as SAM [1] and greenius [2], are widely used within the CSP community to evaluate system configurations, component designs and operating strategies. These existing tools perform well for conventional designs, but can be challenging to customise when researching novel system configurations.

This paper introduces a new CSP simulator, SolarTherm, which provides a platform on which researchers can rapidly prototype unconventional designs and perform flexible yearly performance simulations of full systems including all of the major components. The tool is not limited to CSP systems that produce electricity; solar fuels, solar metallurgy and CST industrial process heat are all intended applications for the tool. Some of the unconventional components that the tool aims to model include particle receivers, solar thermochemical reactors, and phase-change storage, all of which are under evaluation in the Australian Solar Thermal Research Initiative (ASTRI) program.

A key design goal of SolarTherm is the utilisation of free and open source software to make it accessible to a wide range of users and to enable anyone to modify it for their own use cases. This paper provides an overview of the tool and its current state of development, as well as a validation against SAM by comparing experiments on two systems: a generic energy-based model and a physical fluid-based model.

The simulations highlight differences with some undocumented or not well understood SAM components, but for the most part there is a good overall agreement. This helps to validate the fundamentals of SolarTherm, which provides confidence for the results of unconventional designs developed under this framework in the future.

The architecture of the tool is introduced in the first section of this paper, followed by a description and comparison of a published SAM simplified parabolic trough system, and a more sophisticated tower system which represents the reference system of the ASTRI project.

ARCHITECTURE

SolarTherm consists of two core parts:

1. a Modelica library of models for common CSP components and power plant configurations; and
2. a collection of tools for preparing input data, running simulations and analysing results.

The library and tools are designed for use with the free open source OpenModelica [3] framework to simulate the Modelica [4] models. As an alternative, the library models are also compatible with and have been tested in the commercial Dymola [5] framework. As these environments have different levels of support for the Modelica language, care was taken to only make use of the common subset of language and library features.

Users of SolarTherm interact with the models and simulation either through a graphical flowsheet-style tool such as OMEdit/Dymola, or by directly editing model files and running the simulation from the command line. Over time the interfaces may evolve to specialise more to CSP simulation, making it easier for new users to get up to speed and enhancing functionality.

Library

SolarTherm component models are written in Modelica, a declarative modelling language that has a focus on hybrid discrete-continuous physical processes. The dynamic behaviour of modular components (e.g., a molten salt storage tank) are defined in terms of differential-algebraic equations and discrete events. Such components are connected together to form a more complex model of the full power plant. Components in SolarTherm are grouped into categories for source input data, collectors, receivers, storage, power blocks, fluid properties and control.

Capital and maintenance costs are parameterised for the different CSP components, enabling automatic scaling as the system design or size changes. Levelised cost of electricity (LCOE) calculations are performed externally using a tool described in the next section, with future plans to also implement these within the Modelica library.

Tools

The tools included in SolarTherm are for processing input data, analysing results, and for simplifying the process of simulating from the command line. Included are the following scripts written in python:

- `st_simulate` – run an annual performance simulation of a system model (optional parameter sweep) and calculate metrics including LCOE and capacity factor
- `st_optimise` – run parameter optimisation using simple gradient-free algorithms (e.g., the Nelder-Mead method or particle swarm optimisation)
- `st_plotmat` – plot time-series results using matplotlib
- `st_wea_to_mo` – convert typical meteorological year (TMY) weather data to a Modelica suitable format.

Much of this basic functionality can already be found within the OMEdit/Dymola graphical interfaces, while these command line tools are intended to make it easy to automate and repeat annual simulations, and to add new functionality such as sensitivity analysis, interfacing with external tools (e.g., ray tracers) and running simulations on a computer cluster.

Usage and Development

SolarTherm is provided free for use under an open source GPLv2 licence, and is hosted on GitHub at www.github.com/SolarTherm/SolarTherm. The repository provides links to documentation that includes installation instructions, a tutorial and development notes. Documentation for the Modelica models is included within the library code itself, which can be viewed by loading the library into OMEdit/Dymola.

This project takes a continuous integration approach to development, with automated unit and system tests run when changes are made to the repository to help ensure that no regressions or new bugs are introduced. Users are free to make improvements to SolarTherm, add new models for their particular CSP systems of interest, and then to commit these changes back to the main repository to make them available to others.

GENERIC SYSTEM MODELLING

The complexity of models should be chosen to achieve the right balance of accuracy, computational efficiency and modelling effort for the problem at hand. Simpler, faster but less accurate models are appropriate in situations where a short development time is required, where it is necessary to perform many simulations, or where the availability of component data is limited. Included in this are experiments that involve parameter sweeps, parameter optimisation or sensitivity analysis, or when iterating on design. If the focus of the simulation is on one component in particular (e.g., the collector optics or grid dispatch strategy), then it might not be necessary to model the rest of the system in high detail.

This section introduces the GenericSystem model in SolarTherm which is designed to mirror the Generic Solar System (GSS) model in SAM. Wagner and Zhu [6] compare the performance of the GSS model to a more detailed physical model for a trough system, and find that the GSS model provides a strong agreement with the more detailed model on annual performance metrics, with accuracy decreasing on short time scales. Comparing the GenericSystem to the SAM GSS provides some validation of the basics of the SolarTherm framework.

Model Description

The SolarTherm GenericSystem model is an energy-based model that is designed to be flexible enough to model a large range of solar systems with storage. Simple polynomial curves and design point parameters are used to model key component quantities (e.g., efficiencies), which are combined to form a model of the whole system. These polynomials can be fit to real-world experimental data or to existing detailed CFD/FEA simulation results. This section provides an overview of the components, indicating where they differ from the SAM GSS, while the full model equations and parameterisations can be obtained from the SolarTherm source.

A weather component interpolates input TMY3 [7] weather data, and calculates solar positions using the simplified Duffie and Beckman [8] equations. Collector performance is currently modelled with altitude/azimuth optical efficiency tables which are provided to the simulator in a simple ASCII or MATLAB format. This optical efficiency includes cosine error, reflectivity, shading, blocking, attenuation and spillage. The intention is for these optical efficiency tables to be calculated using external ray tracing tools such as Tonatiah [9] for the field/collector design of interest.

Receiver losses are a function of incident heat, ambient dry-bulb temperature and wind speed. The GenericSystem model using a more generic form for this relation than the GSS model, and uses the total heat incident on the receiver instead of solar irradiance as it has a more direct relation to thermal losses.

A simple energy storage tank is used with heat dissipation losses that depend on the charge level and ambient temperature. Heat is dispatched from the tank to the power block which has a cycle efficiency that is a function of temperature and operating point. The net plant output is adjusted to account for a fixed parasitic and a variable parasitic that is a function of ambient temperature and power cycle output.

One major difference from the SAM GSS model is in how the power block startup delay (modelling a warming phase) is handled. The implementation in SAM is unknown, and it is not clear how to interpret the relevant parameters. In the GenericSystem model parameters include: time for the power block to heat from cold, time to cool from hot and the thermal power dispatch required during startup as a percentage of the design point. These parameters were set to 30 min, 60 min and 20% respectively for the trough model used in the experiments.

Operating Strategy

The GenericSystem model has two points of control: collector on/off sun and the dispatch of thermal energy from the storage tank to the power block. The collector control is binary, and is determined by sun altitude deploy and stow angles. The dispatch can be scheduled to operate at different levels depending on the time of day, day of week and month of year, however, for these experiments the control was scheduled to dispatch continuously at the design point.

The scheduled power is only a target, which a lower-level controller takes as an input to determine the true dispatch based on other factors such as the state of the power block and amount of stored energy. The SAM GSS model does not fully discharge the tank in the evening, instead saving enough to enable fast starting in the morning, while the GenericSystem model continues to discharge until the minimum tank level is reached.

Comparison

Parameters from the SAM GSS parabolic trough system developed by Wagner and Zhu [6] were used to configure an equivalent GenericSystem model. The following adjustments were made to the SAM GSS model to enable a more direct comparison: TMY3 data file used instead of TMY2 weather data file, power block maximum overdesign operation set to one and dry bulb reference temperature used for the power block.

The experiments in this section were performed on a quad-core Intel i7-2600 CPU with 8 GB of RAM, using version 1.9.4 of the OpenModelica compiler and version 2015.6.30 of SAM. The Runge-Kutta method (RK4) was used in OpenModelica with fixed 5 min time steps. Commit 593462b of the SolarTherm master branch was used.

Simulating the trough model for one year takes 3.8 s in SolarTherm compared to 200 ms for SAM. Having a more flexible modelling environment is expected to contribute to the slower solve times. Additionally SolarTherm provides access to higher resolution 5 min output data against the 1 h resolution of the SAM output. Further speed improvements might be obtainable by using variable step size solvers such as DASSL, however, in OpenModelica DASSL can cause problems when used with long solve durations (which is a known but unresolved issue).

Table 1 compares annual performance metrics for the trough model in SAM and SolarTherm. The net annual energy produced is 3% higher in the SolarTherm simulation, which is also the cause of the capacity factor and LCOE differences. There are three parts of the model that contribute to these differences: the thermal power incident on the field, the model of receiver losses and the power block startup.

TABLE 1. Annual performance for both SAM and SolarTherm simulations.

Result	SAM	SolarTherm	Diff (%)
Average DNI (kWh/m ² /day)	7.46	7.46	0.00%
Annual Energy (GWh)	371	382	2.95%
Capacity Factor (%)	42.4	43.7	3.07%
LCOE (c/kWh)	14.0	13.6	-2.86%

SAM consistently produces 2.73% higher instantaneous “Field thermal power incident” values from SolarTherm as well as an independent hand calculation of the quantity. There appears to be an undocumented calculation occurring in SAM and we are in the process of contacting NREL to help resolve the difference. The receiver losses are also different, even when we implemented the exact same equations as documented. Either the documentation is incorrect or again we have misinterpreted it. The final difference in the power block startup is not surprising as the equivalent SAM documentation could not be found for comparison.

Figure 1 compares the results of SolarTherm (solid lines) to SAM (dashed lines and dots) for 3 days of the annual simulation. The receiver heat output in Fig. 1(a) shows the effect of the incident field thermal power and the receiver loss model differences. The electrical power and stored energy in Fig. 1(b-c) highlight the differences in the dispatch strategy implementation and the power block startup models. The SAM model attempts to store enough energy overnight so that the power block can start earlier in the morning, but as a result it also runs out of energy earlier in the evening/night.

These results help to validate the SolarTherm framework, not just for this particular Trough model, but also for parts common to other SolarTherm models. The intended differences with the SAM model (such as dispatch strategy) behave as expected, while the unintended differences may only be explored if further documentation of the SAM implementation becomes available. This illustrates one of the key advantages of SolarTherm for the user community: all aspects of the model and solver are visible and can be modified.

Parameter Sweeps

A parameter sweep and simple sensitivity experiment were performed on the trough model to demonstrate these capabilities in SolarTherm. The parameter sweep in Fig. 1(d) shows the influence of solar multiple on LCOE and capacity factor. These results suggest that it would be advantageous to increase the solar multiple from its original value of 1.93. Five parameters were selected for testing the model sensitivity as seen in Table 2. They were each changed by 10% (in the direction which is expected to improve system performance) and simulated, with the results for the LCOE shown in the table. Changing the power cycle efficiency has the greatest impact because it allows the upstream components to be downsized including the storage, receiver and collector.

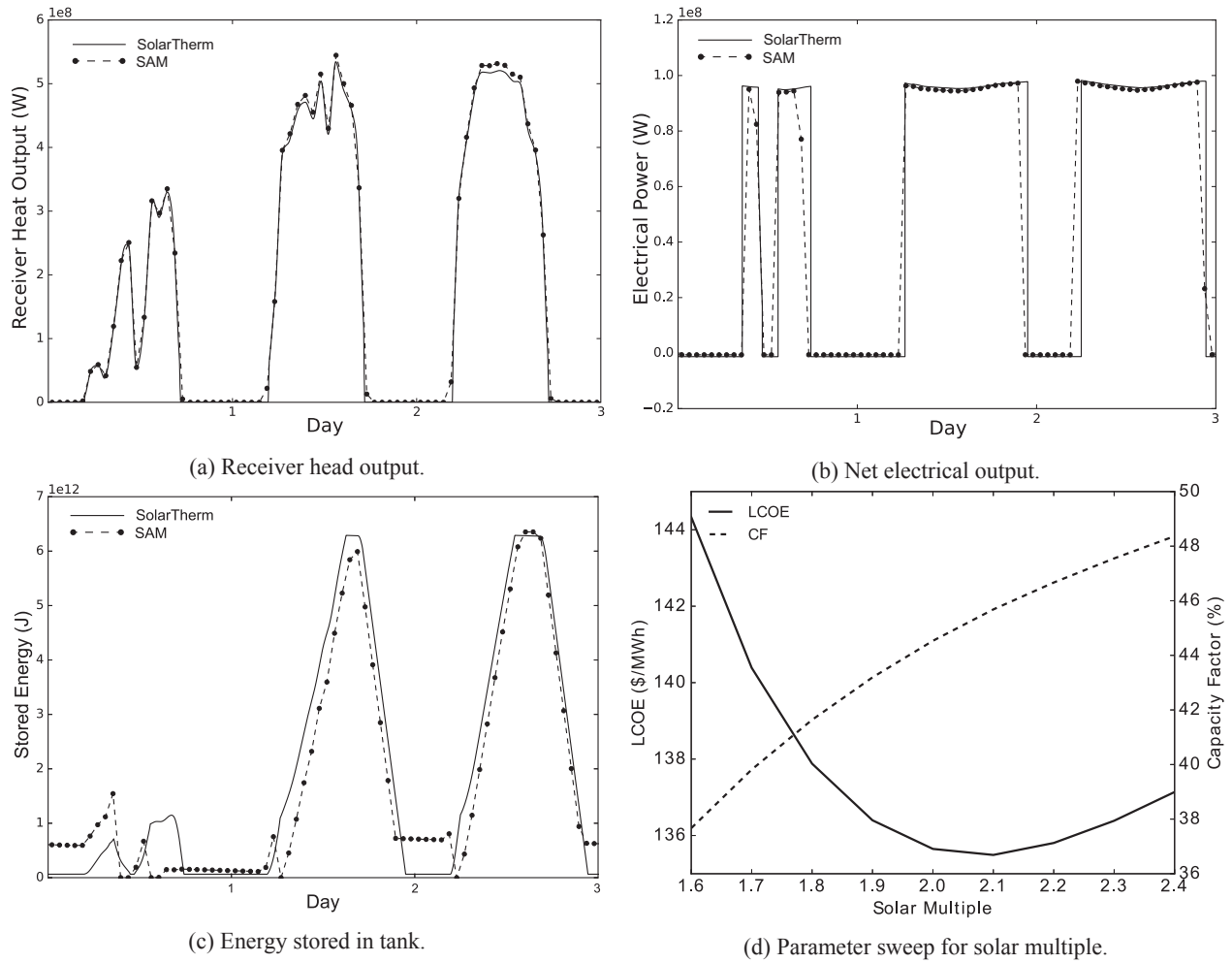


FIGURE 1. GenericSystem comparison to SAM GSS for three consecutive days and parameter sweep result.

TABLE 2. Trough model sensitivity to various model parameters.

Parameter	LCOE Change (%)
Power cycle efficiency	-6.23
Field price	-4.65
Power block price	-1.60
Receiver loss	-0.68
Power block start delay	-0.21

PHYSICAL SYSTEM MODELLING

Although energy-based models can give quite accurate annual performance data, they are unable to predict short-time transient behaviour and have only limited capability to give estimates of the sensitivity of the model to design changes, especially in relation to heat transfer and thermodynamic processes. In such circumstances, models incorporating equations with more detailed physics are required. Physical models involve properties of materials, thermal resistances and empirical equations for heat transfer phenomena. The added physics result in additional equations and variables throughout the model, which can make the model slower to solve, so it is important to carefully select a suitable level of detail.

SolarTherm includes CSP components with physical models that include the main heat and mass transfer processes. In order to be fully compatible with the Modelica Standard Library (MSL), Modelica Fluid and Modelica

Thermal connectors have been used in order to define the relationships between components. The thermodynamic properties have been calculated with medium models that extend the Modelica Media interface.

The library is an ongoing development project that currently contains conventional models for each of the core parts of a CSP plant, based on simple equations that describe the most important phenomena. More complex models (both conventional and unconventional) are expected to be included soon, driven primarily by the modelling requirements within the ASTRI project. Over time the library will continue to evolve with contributions from the wider research community.

Reference Model

In order to test the validity of the library, the ASTRI reference case described by Aghaeimeybodi and Beath [10] has been used as example. This system (shown in Fig. 2), is a typical two-tank molten salt solar tower with a gross capacity of 111 MWe and four hours of thermal storage. At the design point (at noon on the summer solstice, with the reference DNI of 950 W/m^2 , ambient temperature of 43°C and relative humidity of 5.5%), the plant delivers 100 MWe of net electricity to the grid.

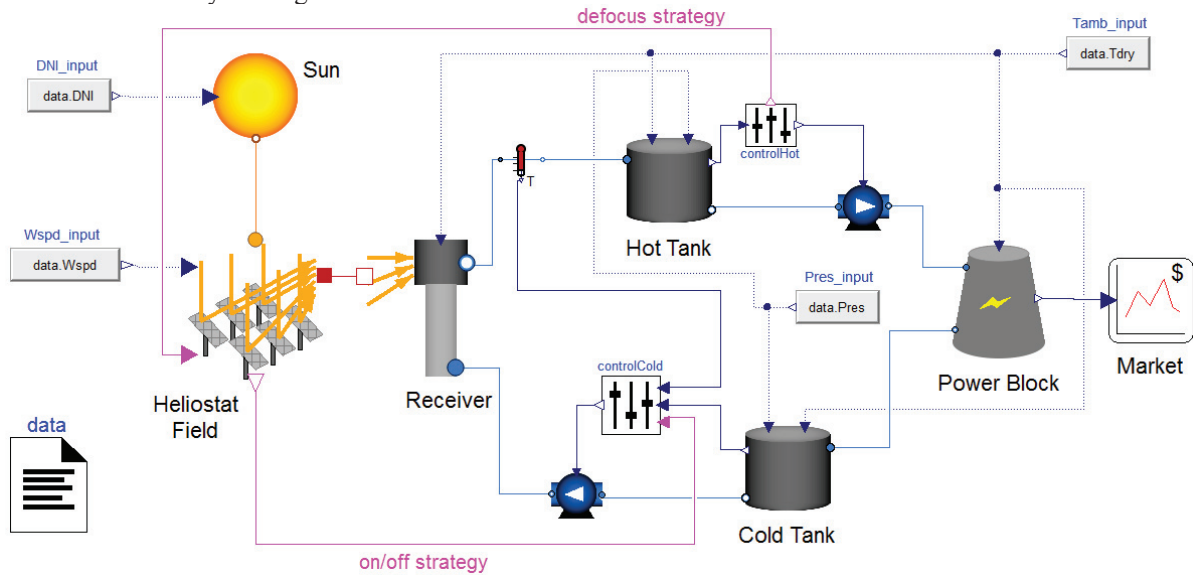


FIGURE 2. The ASTRI reference system as shown from Dymola, which used in the experiments.

The components in the model include: a data source, a sun, a heliostats field, a receiver, two tanks, two pumps, two pump controllers and a power block. A molten salt medium model, whose state variables are the specific enthalpy and the pressure, provides the thermodynamic properties of the working fluid. This working fluid is the common mixture of 60% NaNO_3 and 40% KNO_3 and the properties have been calculated following the results published by Zavoico [11] and Ferri et al. [12] for the liquid region of 290 to 602°C .

The main differences from the generic energy-based model from the previous section are in the fluid circuit and the operating strategy:

Receiver: The fluid entering the receiver is heated as a function of the concentrated solar power, considering radiative and convective heat losses but neglecting effects due to receiver shape and thermal mass.

Tank: Each tank consists of a single control volume where the height and the diameter are design parameters of the model and the top pressure can be a parameter or a variable connected from outside. The model can provide the level of the tank as function of the stored mass and its density through a connector for control purposes. Thermal losses following Newton's law of cooling can be optionally turned on, where the heat transfer coefficient is a parameter and the heat transfer area is a function of the level.

Power Block: The power block replicates the regression model of a Rankine cycle power block [13]. This model calculates the change in the output variables (i.e. electrical power production and heat rejection) with respect to the deviation of the inputs from their design point reference values.

Pump: Ideal pumps are used to drive a controlled mass flow between two points at the same streamline.

Cold Control: This component automates the operating strategy of the pump placed before the receiver. The pump runs when the solar field is working and the level of the cold tank is higher than the minimum level. The mass flow is calculated by a PI controller with anti-windup and reset, which fixes the working fluid receiver outlet temperature to a target value.

Hot Control: This component automates the operating strategy of the pump placed before the power block. The pump discharges the hot tank with a fixed mass flow rate when its level is higher than the start-up level and the discharge carries on while the level is higher than the minimum level. This component also generates a signal for defocusing part of the solar field when the hot tank reaches its maximum level.

Comparison

The physical models were simulated using SAM 2014.1.14, and the Modelica tools Dymola 2017 and OMedit v1.9.5, using a standard laptop (Intel i5 hyper-threading dual-core processor with 8 GB of RAM). The CPU-time for integration for a full year was: 4 s in SAM with an output interval of 1 hour, 18.5 s in Dymola using Lsodar as the numerical solver with an output interval of 60 s and 40.6 s in OMedit using DASSL as the numerical solver with the same output interval.

The Modelica model has 246 scalar equations and 5 continuous-time states, and the simulation has 4270 state events in Dymola and 3867 in OMedit for a full year. Both simulation plots look the same, for this reason only the Dymola one has been shown in Fig. 3. This figure compares the SAM and SolarTherm simulation of the ASTRI reference CSP plant with the same weather data between the 5th and 10th of September.

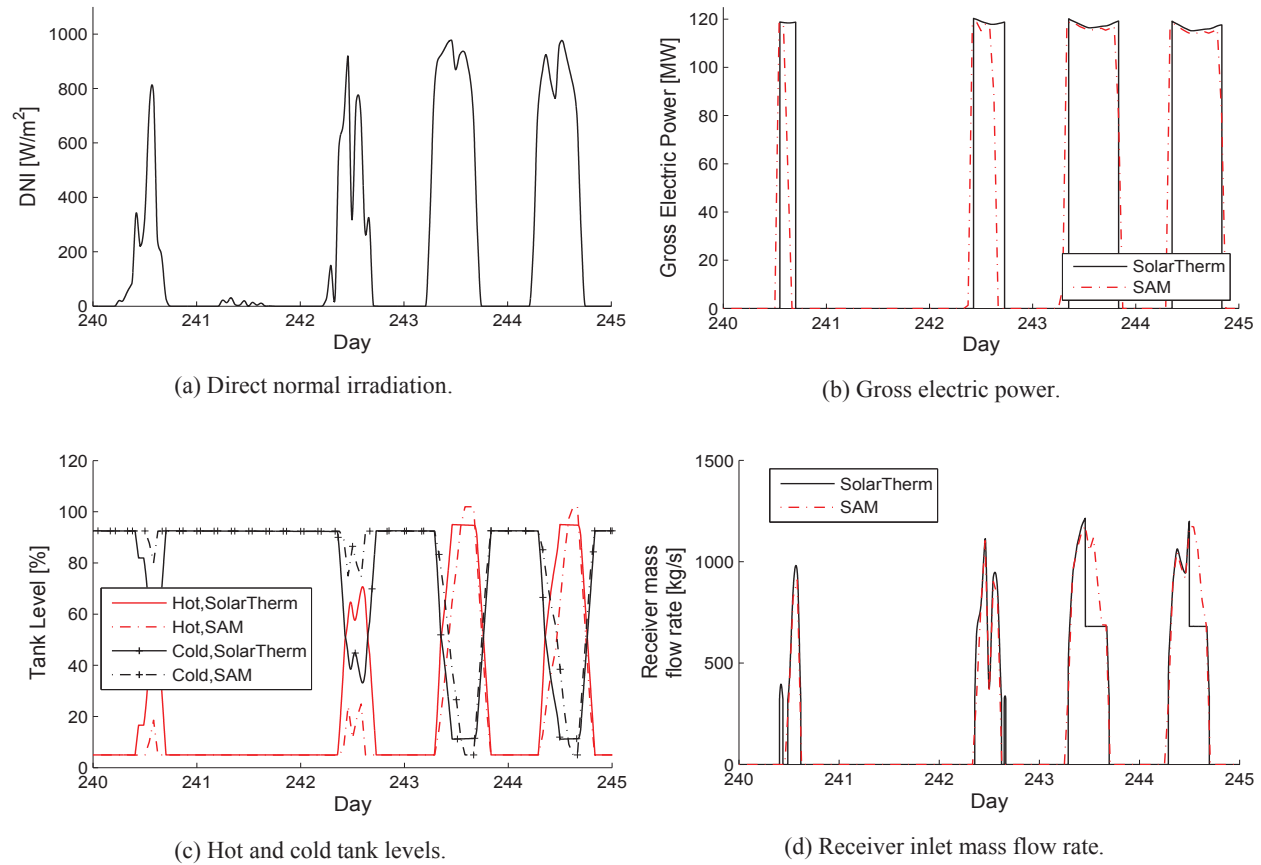


FIGURE 3. Physical model comparison.

Figure 3(a) indicates the variation of the most important input, DNI. Figure 3(b) depicts the gross electric power generated by the plant. Despite the instant power is the same in both models in full-load turbine operation, the annual accumulated power is a 2.89% higher in SolarTherm. The difference in charge/discharge operating strategies seems the origin of this discrepancy. This effect can also be observed in the tank level, see Fig. 3(c). The SolarTherm model allows the plant to continue operation at low DNI values. Another difference is that while SolarTherm impedes the charge/discharge beyond the 5 and 95% of the tank levels, the SAM strategy appears to focus only on the cold tank, allowing it sometimes to fill beyond the maximum storage volume. In SolarTherm a defocusing strategy is activated when the level of the hot tank reaches 95%. The sudden reduction of the receiver inlet mass flow rate is caused by this fact, see Fig. 3(d). Apart from this, both simulations converge to similar values.

CONCLUSIONS

SolarTherm is a new annual performance modelling tool for CSP systems, with both simple energy-based as well as more detailed physical plant models implemented. This paper has validated SolarTherm against SAM for a simple trough model and a physical tower/salt model and although differences exist, it is useful in its current form, especially for research-level evaluation of new concepts.

Future work focus on adapting current SolarTherm models to evaluating a range of novel ASTRI CSP components including PCM storage, sCO₂ power cycles, new heliostat field configurations, sodium and particle receivers, and solar fuels systems. As the code is open-source we are very eager to find new collaborations and contributors.

ACKNOWLEDGEMENTS

This work was performed as part of the ASTRI, a project supported by the Australian Government, through the Australian Renewable Energy Agency (ARENA).

REFERENCES

1. SAM (2016). Solar Advisor Model, <https://sam.nrel.gov/>.
2. greenius (2016). greenius User Manual, Version 4.1, DLR, http://freegreenius.dlr.de/images/greeniusdata/-Greenius-Manual_4_1.pdf.
3. Fritzson, P.; Aronsson, P.; Lundvall, H.; Nyström, K.; Pop, A.; Saldamli, L. and Broman, D. (2005). The OpenModelica Modeling, Simulation, and Software Development Environment, Simulation News Europe 44.
4. The Modelica Association (2016). Modelica, <https://www.modelica.org/>.
5. Dassault Systemes (2016). CATIA Systems Engineering --- Dymola, <http://www.3ds.com/products-services/catia/products/dymola/>.
6. Wagner, M. and Zhu, G. (2011). A Generic CSP Performance Model for NREL's System Advisor Model. SolarPACES 2011.
7. Wilcox, S. and Marion, W. (2008). Users Manual for TMY3 Data Sets, NREL.
8. Duffie, J. and Beckman, W., 2013. Solar Engineering of Thermal Processes. Wiley.
9. Blanco, M. J.; Amieva, J. M. and Mancillas, A. (2005). The Tonatiuh Software Development Project: An open source approach to the simulation of solar concentrating systems. ASME 2005 International Mechanical Engineering Congress and Exposition.
10. Aghaeimeybodi, M. and Beath, A. (2015). Central receiver reference plant technical and cost data, Internal report: 111.
11. Zavoico, A. B. (2001). Solar Power Tower - Design Basis Document: 148.
12. Ferri, R.; Cammi, A. and Mazzei, D. (2008). Molten salt mixture properties in RELAP5 code for thermodynamic solar applications, *International Journal of Thermal Sciences* 47 : 1676-1687.
13. Wagner, M. J. (2008). Simulation and Predictive Performance Modeling of Utility-Scale Central Receiver System Power Plants, University of Wisconsin - Madison.