

## *Research Article*

# **SDSim: A Novel Simulator for Solar Drying Processes**

**Yolanda Bolea, Antoni Grau, and Alexandre Miranda**

*Automatic Control Department, Technical University of Catalonia UPC, E-08028 Barcelona, Spain*

Correspondence should be addressed to Antoni Grau, antoni.grau@upc.edu

Received 16 March 2012; Accepted 26 April 2012

Academic Editor: Zhijun Zhang

Copyright © 2012 Yolanda Bolea et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

SDSim is a novel solar dryer simulator based in a multicrop, inclined multipass solar air heater with in-built thermal storage mathematical model. This model has been developed as a designing and developing tool used to study and forecast the behavior of the system model in order to improve its drying efficiency and achieving a return on the dryer investment. The main feature of this simulator is that most of the parameters are permitted to be changed during the simulation process allowing finding the more suitable system for any specific situation with a user-friendly environment. The model has been evaluated in a real solar dryer system by comparing model estimates to collected data.

## **1. Introduction**

Food drying is a very simple, ancient skill. It is one of the most accessible and hence the most widespread processing technology. Sun drying of fruits and vegetables is still practiced largely unchanged from ancient times. Traditional sun drying takes place by storing the product under direct sunlight. Sun drying is only possible in areas where, in an average year, the weather allows foods to be dried immediately after harvest. The main advantages of sun drying are low capital and operating costs and the fact that little expertise is required. The main disadvantages of this method are as follows: contamination, theft, or damage by birds, rats, or insects; slow or intermittent drying and no protection from rain or dew that wets the product encourages mould growth and may result in a relatively high final moisture content; low and variable quality of products due to over- or underdrying; large areas of land needed for the shallow layers of food; laborious since the crop must be turned, moved if it rains; direct exposure to sunlight reduces the quality (color and vitamin content) of some fruits and vegetables. Moreover, since sun drying depends on uncontrolled factors, production of uniform and standard products is not expected. The quality of sun-dried foods can be

improved by reducing the size of pieces to achieve faster drying and by drying on raised platforms, covered with cloth or netting to protect against insects and animals [1, 2].

Due to the current trends towards higher cost of fossil fuels and uncertainty regarding future cost and availability, the use of solar energy in food processing will probably increase and become more economically feasible in the near future. Solar dryers have some advantages over sun drying when correctly designed [3–7]. They give faster drying rates by heating the air to 10–30°C above ambient, which causes the air to move faster through the dryer, reduces its humidity, and deters insects. The faster drying reduces the risk of spoilage, improves quality of the product and gives a higher throughput, so reducing the drying area that is needed. However care is needed when drying fruits to prevent too rapid drying which will prevent complete drying and would result in case hardening and subsequent mould growth. Solar dryers also protect foods from dust, insects, birds, and animals. They can be constructed from locally available materials at a relatively low capital cost, and there are no fuel costs. Thus, they can be useful in areas where fuel or electricity are expensive, land for sun drying is in short supply or expensive, sunshine is plentiful, but the air humidity is high. Moreover, they may be useful as a means of heating air for artificial dryers to reduce fuel costs [8]. Solar food drying can be used in most areas but how quickly the food dries is affected by many variables, especially the amount of sunlight and relative humidity. Typical drying times in solar dryers range from 1 to 3 days depending on sun, air movement, humidity, and the type of food to be dried [9, 10].

The principle that lies behind the design of solar dryers is as follows: in drying relative and absolute humidity are of great importance. Air can take up moisture, but only up to a limit. This limit is the absolute (maximum) humidity, and it is temperature dependent. When air passes over a moist food, it will take up moisture until it is virtually fully saturated, that is, until absolute humidity has been reached. But, the capacity of the air for taking up this moisture is dependent on its temperature. The higher the temperature, the higher the absolute humidity is and the larger the uptake of moisture is too. If air is warmed, the amount of moisture in it remains the same, but the relative humidity falls, and the air is therefore enabled to take up more moisture from its surrounding. To produce a high-quality product economically, it must be dried fast, but without using excessive heat, which could cause product degradation. Drying time can be shortened by two main procedures; one is to raise the product temperature so that the moisture can be readily vaporized, while at the same time the humid air is constantly being removed. The second is to treat the product to be dried so that the moisture barriers, such as dense hydrophobic skin layers or long water migration paths, will be minimized [11].

In environmental and agricultural sciences, complex systems need often to be described with mathematical models. The development of model structures adequate for practical use is carried out with different approaches, depending on the goals of the modeling process as well as on the available information. Most part of environmental and agricultural processes are intrinsically distributed parameter systems, and their behavior is therefore naturally described by partial differential equations (PDEs) that, besides being function of time, depend also on spatial coordinates. Possible examples are given by processes in which mass or energy transport phenomena occur. The resulting models are infinite-dimensional state models. Usually this kind of equations does not have analytical solutions, and numerical methods (such as, Euler method, finite elements techniques, Preissman method, characteristics methods, etc.) are used for their resolution. This type of description usually involves a huge number of parameters and requires time-consuming computations [12]. In the case of solar crop drying system involves the transport of moisture to the surface

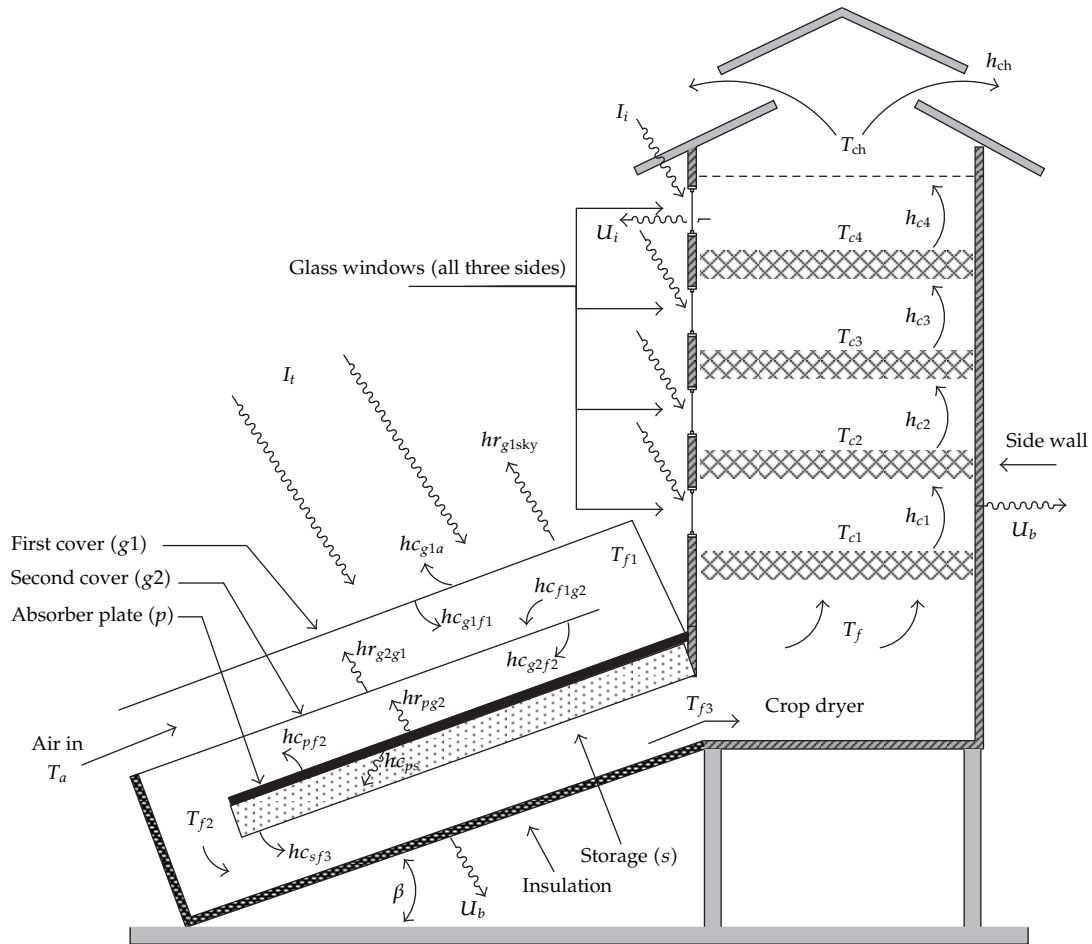
of the product and subsequent evaporation of the moisture by thermal heating. Thus, solar thermal crop-drying is a complex process of simultaneous heat and mass transfer. Then, to represent of behavior of the crop drying plant of a precise and complete way, a partial differential equations system is needed.

Agricultural models and decision support systems (DSSs) are becoming increasingly available for a wide audience of users. The Great Plains Framework for Agricultural Resource Management (GPFARM) DSS is a strategic planning tool for farmers, ranchers, and agricultural consultants that incorporates a science simulation model with an economic analysis package and multicriteria decision aid for evaluating individual fields or aggregating to the entire enterprise. The GPFARM DSS is currently being expanded to include (1) better strategic planning by simulating a greater range of crops over a wider geographic range and management systems, (2) incorporating a tactical planning component, and (3) adding a production, environmental, and economic risk component [13]. Another farming simulator in this case specifically for harvesting systems in agriculture is "CropSyst", that is, a multiyear, multicrop, daily time step cropping systems simulation model developed to serve as an analytical tool to study the effect of climate, soils, and management on cropping systems productivity and the environment. The development of this simulator started in the early 1990s, evolving to a suite of programs including a cropping systems simulator, a weather generator (ClimGen), GIS-CropSyst cooperater program (ArcCs), a watershed model (CropSyst Watershed), and several miscellaneous utility programs [14]. Specifically in this simulator a mathematical model which is intended for crop growth simulation over a unit field area ( $m^2$ ) is implemented [15].

Nowadays in farming sector one of the most potential future applications is the solar drying of agricultural products. Losses of fruits and vegetables during their drying in developing countries are estimated to be 30–40% of production. The postharvest losses of agricultural products in the rural areas of the developing countries can be reduced drastically by using well-designed solar drying systems. Before using the drying systems on large scale, computer simulation models must be performed to simulate the short and long-term performance of the drying systems with and without the storage media to estimate the solar drying curves of the dried products and investigate the cost benefits of the solar drying of agricultural products. So far, some related works [16–23] perform the study and analysis about this kind of systems using the modeling and evolution of their behavior computed mainly in MATLAB (but also in other proprietary code) by the research authors. Therefore, in solar drying systems research as well as in their management, the engineers, researchers, and managers are required to implement their own model to simulate the drying behavior prior to their implementation. This fact is a drawback because there do not exist open source simulators or interactive software to make easier the task of simulating the systems with changing parameters or specifications. The simulator that authors present in this paper is intended to implement a generic user-friendly crop-drying simulator based on solar energy that will reproduce the system behavior and will be accessible to any user requiring a study about this problem without the need of creating his/her own simulator-wasting time and resources. The mathematical model that has been implemented in the simulator is based in a multitray crop drying using inclined multipass solar air heater with in-built thermal storage.

Considering the importance of solar crop drying, this new simulator (SDSim) has been implemented with Easy Java Simulation (Ejs). This software developing tool is explained and referenced in Section 3.

Using the SDSim simulator the temperatures of the different trays, the moisture evaporation, and the drying rate for each produce are predicted. Also the dryer efficiency



**Figure 1:** Multitray crop dryer with inclined multipass air heater with in-built thermal storage showing the distribution of solar energy and various heat transfer coefficients.

is another important value evaluated after each simulation allowing the engineer to choose the most suitable design for each situation.

In Section 2 the model is explained in detail. In Section 3 the simulation software is reviewed. Paper finishes with some interesting results and conclusions.

## 2. Thermal Mathematical Modeling

The implemented equations in the simulator are generic, and for this reason the user can reproduce the behavior of any multitray crop drying system with inclined multipass solar air heater with in-built thermal storage simply varying its geometry and features. For the sake of simplicity, in order to evaluate and validate this simulator, a real test bench multicrop drying system and its corresponding equations will be used. The specific features of this system are described as in the nomenclature section.

This system consists on a multitray dryer fitted with a solar air heater (SAH) developed in India [19]. A schematic diagram is shown in Figure 1. The system is assumed

to face towards the midday sun. The solar air heater is oriented to face south and tilted at  $\beta^\circ$  angle from the horizontal plane. The system has four drying trays at equal vertical spacing. The drying chamber is provided with glass windows on three sides, that is, east, south, and west to receive additional solar energy. The rear side (north) is provided with an insulating wall. The dryer receives energy from the solar air heater at the first tray. Part of this heat is used to dry the crop in the first tray, and the rest is transferred to the second, third, and fourth trays. Solar energy entering through the windows is absorbed by the crop surface, heats up the crop and accelerates the drying rate. Thus, it improves the system performance. The whole system works as a mixed-mode (passive and active) dryer.

This process involves the mass and heat transfer. Therefore, dried food and the glass' properties should be studied, as the thermodynamic properties. The solar radiation is transmitted from the glass covers and is absorbed by the absorber plate. The air flows in between the glass covers, above the absorber plate, and below the storage material, where it is heated. The energy balance equations on the various components of the system are written with the following assumptions: (i) the heat capacities of the air, glass cover, absorber plate, and insulation are negligible, (ii) there is no temperature gradient along the thickness of the glass cover, (iii) storage material has an average temperature ( $T_s$ ) at a time ( $t$ ), (this assumption may be achieved with the small thickness of storage material), (iv) no stratification exists perpendicular to the air flow in the ducts, (v) the system is perfectly insulated, and there is no air leakage, (vi) the volume shrinkage of the crop is negligible during drying, (vii) the system is facing towards midday sun.

The mathematical model that describes the solar drier behavior is represented by the following laws.

## 2.1. Energy Balance on Solar Air Heater

### *First Glass Cover*

We have

$$\begin{aligned} \alpha_{g1} \cdot I_t \cdot A_{g1} + hr_{g2g1} \cdot (T_{g2} - T_{g1}) \cdot A_{g1} \\ = hc_{g1f1} \cdot (T_{g1} - T_{f1}) \cdot A_{g1} + hc_{g1a} \cdot (T_{g1} - T_a) \cdot A_{g1} \\ + hr_{g1sky} (T_{g1} - T_{sky}) \cdot A_{g1}, \end{aligned} \quad (2.1)$$

where  $T_{sky} = T_a - 6$  [24].

### *Second Glass Cover*

We have

$$\begin{aligned} \tau_{g2} \cdot \alpha_{g2} \cdot I_t \cdot A_{g2} + hr_{pg2} (T_p - T_{g2}) \cdot A_{g2} + hc_{f1g2} \cdot (T_{f1} - T_{g2}) \cdot A_{g2} \\ = hr_{g2g1} \cdot (T_{g2} - T_{g1}) \cdot A_{g2} + hc_{g2f2} \cdot (T_{g2} - T_{f2}) \cdot A_{g2} \end{aligned} \quad (2.2)$$

*Absorber Plate*

We have

$$\tau_{g1} \cdot \tau_{g2} \cdot \alpha_p \cdot I_t \cdot A_p = hr_{pg2} \cdot (T_p - T_{g2}) \cdot A_p + hc_{pf2} \cdot (T_p - T_{f2}) \cdot A_p + hc_{ps} \cdot (T_p - T_s) \cdot A_p. \quad (2.3)$$

*Fluid Entrance during the First Glass Cover (for Air Stream I)*

We have

$$hc_{g1f1} \cdot (T_{g1} - T_{f1}) \cdot bdx = \dot{m}_a \cdot C_a \cdot \frac{dT_{f1}}{dx} \cdot dx + hc_{f1g2} \cdot (T_{f1} - T_{g2}) \cdot bdx. \quad (2.4)$$

*Fluid during the Second Glass Cover (for Air Stream II)*

One has

$$hc_{g2f2} \cdot (T_{g2} - T_{f2}) \cdot bdx + hc_{pf2} \cdot (T_p - T_{f2}) \cdot bdx = \dot{m}_a \cdot C_a \cdot \frac{dT_{f2}}{dx} \cdot dx. \quad (2.5)$$

*Fluid during the Third Glass Cover (for Air Stream III)*

One has

$$hc_{sf3} \cdot (T_s - T_{f3}) \cdot bdx = \dot{m}_a \cdot C_a \cdot \frac{dT_{f3}}{dx} \cdot dx + Ub \cdot (T_{f3} - T_a) \cdot bdx. \quad (2.6)$$

## **2.2. Storage Material**

One has

$$hc_{ps} \cdot (T_p - T_s) \cdot A_p = \dot{m}_s \cdot C_s \cdot \frac{dT_s}{dt} \cdot A_p \cdot hc_{sf3} (T_s - T_{f3}). \quad (2.7)$$

*Energy Outlet from Solar Air Heater and Energy Balance on Drying Cabin*

Useful energy from solar air heater [25]

$$\dot{Q}_u = \dot{m}_a \cdot C_a \cdot (T_f - T_a) \quad (2.8)$$

### 2.3. Energy Balance on Different Trays in the Drying Chamber

#### First Tray

We have

$$\begin{aligned} \dot{Q}_u + \alpha_c \cdot \sum I_i \cdot A_{w1i} \cdot \tau_i - \left( \sum U_i \cdot A_{w1i} + U_b \cdot A_{w1b} \right) \cdot (T_{c1} - T_a) \\ = M_{c1} \cdot C_c \cdot \frac{dT_{c1}}{dt} + A_c \cdot h_{c1} \cdot (T_{c1} - T_{c2}). \end{aligned} \quad (2.9)$$

#### Second Tray

We have

$$\begin{aligned} A_c \cdot h_{c1} \cdot (T_{c1} - T_{c2}) + \alpha_c \cdot \sum I_i \cdot A_{w2i} \cdot \tau_i - \left( \sum U_i \cdot A_{w2i} + U_b \cdot A_{w2b} \right) \cdot (T_{c2} - T_a) \\ = M_{c2} \cdot C_c \cdot \frac{dT_{c2}}{dt} + A_c \cdot h_{c2} \cdot (T_{c2} - T_{c3}). \end{aligned} \quad (2.10)$$

#### Third Tray

One has

$$\begin{aligned} A_c \cdot h_{c2} \cdot (T_{c2} - T_{c3}) + \alpha_c \cdot \sum I_i \cdot A_{w3i} \cdot \tau_i - \left( \sum U_i \cdot A_{w3i} + U_b \cdot A_{w3b} \right) \cdot (T_{c3} - T_a) \\ = M_{c3} \cdot C_c \cdot \frac{dT_{c3}}{dt} + A_c \cdot h_{c3} \cdot (T_{c3} - T_{c4}). \end{aligned} \quad (2.11)$$

#### Fourth Tray

We have

$$\begin{aligned} A_c \cdot h_{c3} \cdot (T_{c3} - T_{c4}) + \alpha_c \cdot \sum I_i \cdot A_{w4i} \cdot \tau_i - \left( \sum U_i \cdot A_{w4i} + U_b \cdot A_{w4b} \right) \cdot (T_{c4} - T_a) \\ = M_{c4} \cdot C_c \cdot \frac{dT_{c4}}{dt} + A_c \cdot h_{c4} \cdot (T_{c4} - T_{ch}). \end{aligned} \quad (2.12)$$

#### Chamber

We have

$$A_c \cdot h_{c4} \cdot (T_{c4} - T_{ch}) = \dot{m}_a \cdot C_f \cdot (T_{ch} - T_a) + h_{ch} \cdot A_{ch} \cdot (T_{ch} - T_a). \quad (2.13)$$

The performance of the solar collector can be evaluated if the crop temperatures ( $T_{c1}$ ,  $T_{c2}$ ,  $T_{c3}$ , and  $T_{c4}$ ) in the different trays are known in above equations. To find out the crop

temperatures, (2.8)–(2.12) can be rearranged into the following four coupled differential equation:

$$\begin{aligned}
 \frac{dT_{c1}}{dt} + k_{11} \cdot T_{c1} + k_{12} \cdot T_{c2} + k_{13} \cdot T_{c3} + k_{14} \cdot T_{c4} &= F_1(t), \\
 \frac{dT_{c2}}{dt} + k_{21} \cdot T_{c1} + k_{22} \cdot T_{c2} + k_{23} \cdot T_{c3} + k_{24} \cdot T_{c4} &= F_2(t), \\
 \frac{dT_{c3}}{dt} + k_{31} \cdot T_{c1} + k_{32} \cdot T_{c2} + k_{33} \cdot T_{c3} + k_{34} \cdot T_{c4} &= F_3(t), \\
 \frac{dT_{c4}}{dt} + k_{41} \cdot T_{c1} + k_{42} \cdot T_{c2} + k_{43} \cdot T_{c3} + k_{44} \cdot T_{c4} &= F_4(t),
 \end{aligned} \tag{2.14}$$

and the coefficients are:

$$\begin{aligned}
 k_{11} &= \frac{h_{c1} \cdot A_c + \sum U_i \cdot A_{w1i} + U_b \cdot A_{w1b}}{M_{c1} \cdot C_c}, & k_{12} &= \frac{-h_{c1} \cdot A_c}{M_{c1} \cdot C_c}, & k_{13} &= 0.0, & k_{14} &= 0.0, \\
 F_1(t) &= \frac{\dot{Q}_u \cdot \alpha_c \cdot \sum I_i \cdot A_{w1i} \tau_i \cdot T_a \sum (U_i \cdot A_{w1i} + U_b \cdot A_{w1b})}{M_{c1} \cdot C_c}, \\
 k_{21} &= \frac{-h_{c1} \cdot A_c}{M_{c2} \cdot C_c}, & k_{22} &= \frac{h_{c2} \cdot A_c + \sum U_i \cdot A_{w2i} + U_b \cdot A_{w2b} + h_{c1} \cdot A_c}{M_{c2} \cdot C_c}, \\
 k_{23} &= \frac{-h_{c2} \cdot A_c}{M_{c2} \cdot C_c}, & k_{24} &= 0.0, \\
 F_2(t) &= \frac{\alpha_c \cdot \sum I_i \cdot A_{w2i} \tau_i + T_a \cdot \sum (U_i \cdot A_{w2i} + U_b \cdot A_{w2b})}{M_{c2} \cdot C_c}, \\
 k_{31} &= 0.0 & k_{32} &= \frac{-h_{c2} \cdot A_c}{M_{c3} \cdot C_c}, \\
 k_{33} &= \frac{h_{c3} \cdot A_c + \sum U_i \cdot A_{w3i} + U_b \cdot A_{w3b} + h_{c2} \cdot A_c}{M_{c3} \cdot C_c}, & k_{34} &= \frac{-h_{c3} \cdot A_c}{M_{c3} \cdot C_c}, \\
 F_3(t) &= \frac{\alpha_c \cdot \sum I_i \cdot A_{w3i} \tau_i \cdot T_a \cdot \sum (U_i \cdot A_{w3i} + U_b \cdot A_{w3b})}{M_{c3} \cdot C_c}, \\
 k_{41} &= 0.0, & k_{42} &= 0.0, & k_{43} &= \frac{-h_{c3} \cdot A_c}{M_{c4} \cdot C_c}, \\
 k_{44} &= \frac{U_{ch} + \sum U_i \cdot A_{w4i} + U_b \cdot A_{w4b} + h_{c3} \cdot A_c}{M_{c4} \cdot C_c}, \\
 F_4(t) &= \frac{\alpha_c \cdot \sum I_i \cdot A_{w4i} \tau_i \cdot T_a \cdot \sum (U_i \cdot A_{w4i} + U_b \cdot A_{w4b}) + T_a \cdot U_{ch}}{M_{c4} \cdot C_c}, \\
 U_{ch} &= \frac{(\dot{m}_a \cdot C_f + h_{ch} \cdot A_{ch}) h_{c4} \cdot A_c}{M_{c4} \cdot C_f + h_{ch} \cdot A_{ch} + h_{c4} \cdot A_c}.
 \end{aligned} \tag{2.15}$$



The rate of moisture content change in thin drying bed can be written as the following drying equation:

$$\frac{dM}{dt} = -K_d(M - M_e), \quad (2.16)$$

where  $K_d = a \exp(-b'/T_c)$ , and  $M_e$  is the equilibrium moisture content. After a considerable time, the evolution of  $M$  tends to  $M_e$ . In this case,  $M_e$  is not a constant value, but it depends on the temperature and the water activity reaching an equilibrium with its variation. Regarding this variable, there exist different models to calculate it corresponding to the family of the desorption isotherms. For an extensive review and models discussion, see [26]. The most accepted model has been proposed by Henderson at [27], and the model is

$$M_e = \left[ -\frac{\ln(1 - a_w)}{A(T_C + B)} \right]^{1/D}. \quad (2.17)$$

The constants and the value for the water activity are found by experimentation in every crop to be dried.

Another important output variable is the system drying efficiency, and it can be understood by the overall thermal efficiency of the drying. This overall thermal efficiency can be defined as the ratio of heat energy used in the vaporization of the moisture plus the thermal energy used to raise the temperature of the crops, to that of solar radiation collected by the solar air heater and the crop surface. In this work, the following expression is used to evaluate it:

$$\eta_o = \frac{L_c \cdot \sum_{t=1}^{24} \sum_{j=1}^4 M_{ev,t,j} + C_c \cdot \sum_{t=1}^{24} \sum_{j=1}^4 M_{c,t,j} \cdot (T_{c,t,j} - T_{c,t-1,j})}{3600 \cdot \sum_{t=1}^{24} (A_p \cdot I_t + A_c \cdot I_i)}, \quad (2.18)$$

where the vertical radiation  $I_i$  is the sum of the radiations that receive the solar drier from the windows oriented to the sun during all the day, that is,

$$I_i = I_{\text{south}} + I_{\text{west}} + I_{\text{east}} \quad (2.19)$$

calculated using classical formulation found in [28, 29].

$M_{ev}$  is the moisture evaporation in kg water, in other words, the variation of the mass of water in the crop. This mass of water can be calculated using the initial value of the mass of the crop (mass of water + mass of dry matter) and its moisture content using the next relations

$$\begin{aligned} \text{mass dry matter} &= \frac{M_c(t=0)}{1 + M(t=0)}, \\ \text{mass of water}(t) &= M(t) \cdot \text{mass dry matter}, \\ M_{ev}(t) &= \text{mass of water}(t-1) - \text{mass of water}(t) \end{aligned} \quad (2.20)$$

evaluated every hour.

### **3. Simulator Development**

#### **3.1. Easy Java Simulations**

Easy Java simulations (EJSs) is the tool that authors chose to program models and the simulation views. EJS is a freeware, open-source tool developed in Java, specifically designed to create interactive dynamic simulations [30]. It was originally designed to be used by students for interactive learning under the supervision of educators with a low programming level. However, the user needs to know in detail the analytical model of the process and the design of the graphical view. EJS guides the user in the process of creating interactive simulations, in a simple and practical way.

The architecture of EJS derives from the model-view control paradigm, whose philosophy is that interactive simulations must be composed of three parts: the model, the view, and the control. According to that, the steps to build an application in EJS are the following. (1) To define the model is necessary to specify the variables that describe the system and the mathematical equations interrelating them, (2) to define the view in order to represent the states of the process, and finally (3) to define the control in order to describe the actions that the modeler can execute above the simulation. These three parts are interconnected because the model affects to the view, and the control actions affect to behavior of the model. Finally, the view affects to the model and to the control because the graphical interface can contain information about them.

EJS simulations are created through specifying a model to be run by the internal simulation engine and by building a view to visualize the model state and that readily responds to user interactions. So, to define the model in EJS, it is necessary to identify the variables that describe the process, to initialize them, and also to describe the mathematical equations that generate the model. The view is the user-to-model interface of interactive simulations. It is intended to provide a visual representation of the model's relevant properties and dynamic behavior and also to facilitate the user's interactive actions. EJS includes a set of ready-to-use visual elements. With them, the modeler can compose a sophisticated view in a simple, drag, and drop way. The properties of the view elements can be linked to the model variables, producing a bi-directional flow of information between the view and the model. Any change of a model variable is automatically displayed by the view. Reciprocally, any user interaction with the view automatically modifies the value of the corresponding model variable. Once the modeler has defined the model and the view of the interactive simulation, EJS generates the Java source code of the simulation program, compiles the program, packs the resulting object files into a compressed file, and generates HTML pages containing the narrative and the simulation as an applet. The user can readily run the simulation and/or publish it on the Internet. Easy Java simulations, the software tool, a complete English manual for it, can be downloaded for free from EJS' web server at [31].

To summarize, the model is the scientific part of the simulation; yet the creation of the necessary graphical user interface (the view) is the part of the simulation that demands more knowledge of advanced programming techniques.

#### **3.2. SDSim Simulator Development**

When developing a simulation with EJSs there are three separate parts: description, model, and views.

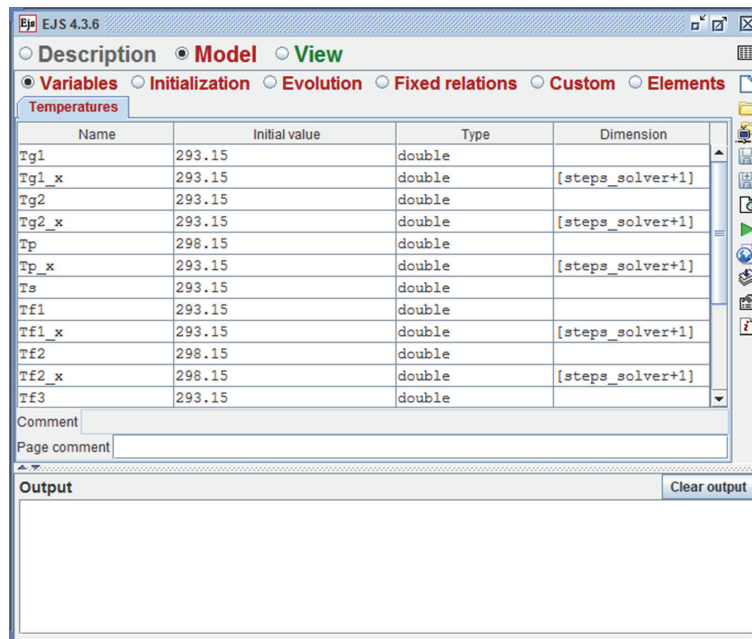


Figure 2: Variables declaration screen.

The description part is devoted to describe the model, as an introduction, with the most relevant equations, parameters and as many figures and pictures as the developer would like to add to make attractive the read of this introduction.

The model part is the heart of the simulator, and only Java code is allowed. All the evolution equations will be programmed as well as their relationships. All the variables and their initializations will be programmed. There is also an Ordinary Differential Equations (ODE) editor where the equations can be programmed.

The view part is necessary to have interaction with the simulation. The view editor allows the programmer to design a layout of the input/output. With some drawing elements, different views can be created and, the most important thing, those drawings elements can be associated to model variables. Then, at simulation time, as the variables change with time, the elements also change creating a dynamic effect of the simulation. Obviously, some plots can also be created. The parameters of the simulation can be also modified at execution time through button or sliders, modifying in real time the simulation. This interaction with the user makes EJS very attractive to develop and utilize this software to design, implement, and use it.

### 3.2.1. The Model

The first step when programming the model is to declare the variables and constants that will be used along the program. Figure 2 shows the variable declaration screen. For any variable, its name, initial value, type, and dimension can be defined.

Once the variables have been declared, if they need to be initialized using not single values but equations or more complex functions, the Initialization tab can be used. All the

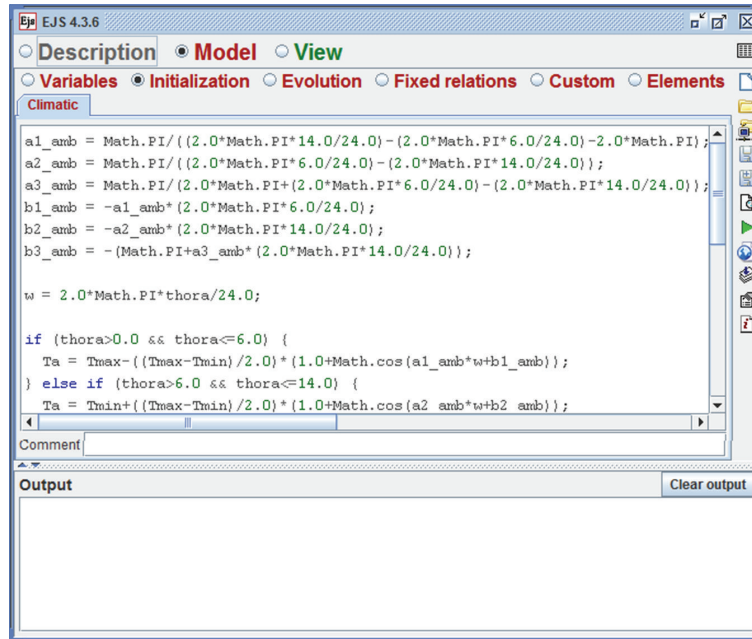


Figure 3: Complex variable initialization.

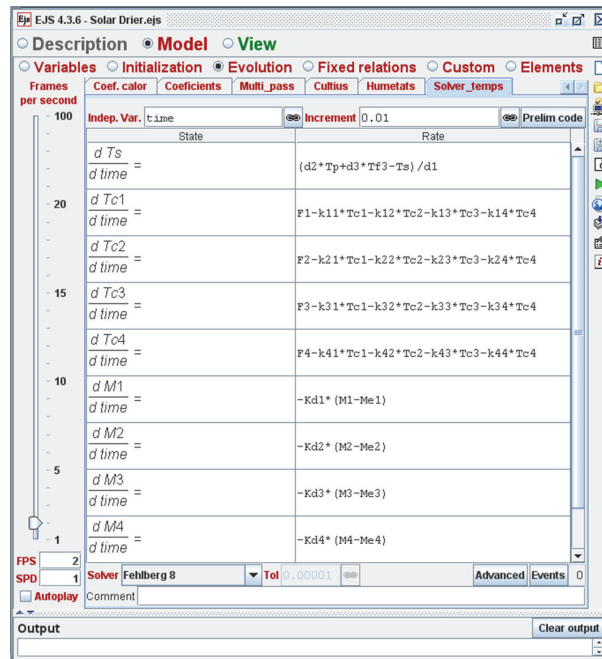
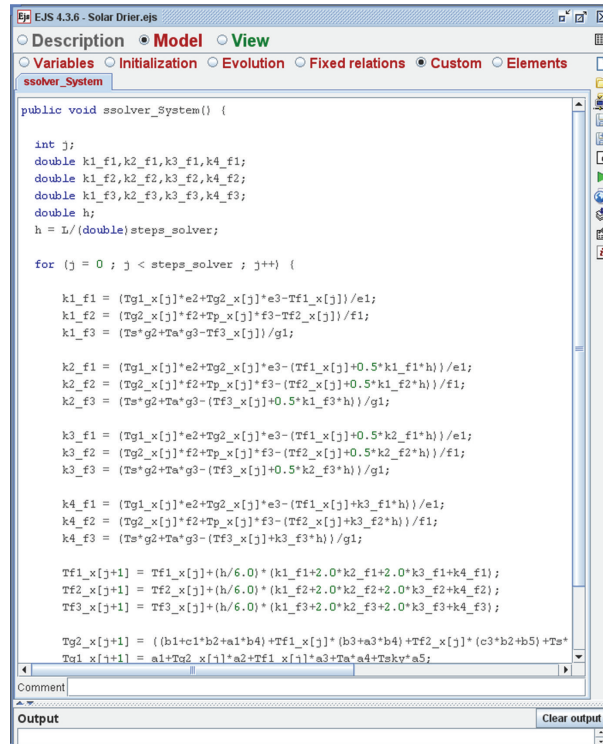


Figure 4: Ordinary differential equations editor.



```

public void ssolver_System() {
    int j;
    double k1_f1,k2_f1,k3_f1,k4_f1;
    double k1_f2,k2_f2,k3_f2,k4_f2;
    double k1_f3,k2_f3,k3_f3,k4_f3;
    double h;
    h = L/(double)steps_solver;

    for (j = 0 ; j < steps_solver ; j++) {

        k1_f1 = (Tg1_x[j]*e2+Tg2_x[j]*e3-Tf1_x[j])/e1;
        k1_f2 = (Tg2_x[j]*f2+Tp_x[j]*f3-(Tf2_x[j]+0.5*k1_f2*h))/f1;
        k1_f3 = (Ts*g2+Ta*g3-Tf3_x[j])/g1;

        k2_f1 = (Tg1_x[j]*e2+Tg2_x[j]*e3-(Tf1_x[j]+0.5*k1_f1*h))/e1;
        k2_f2 = (Tg2_x[j]*f2+Tp_x[j]*f3-(Tf2_x[j]+0.5*k2_f2*h))/f1;
        k2_f3 = (Ts*g2+Ta*g3-(Tf3_x[j]+0.5*k2_f3*h))/g1;

        k3_f1 = (Tg1_x[j]*e2+Tg2_x[j]*e3-(Tf1_x[j]+0.5*k2_f1*h))/e1;
        k3_f2 = (Tg2_x[j]*f2+Tp_x[j]*f3-(Tf2_x[j]+0.5*k2_f2*h))/f1;
        k3_f3 = (Ts*g2+Ta*g3-(Tf3_x[j]+0.5*k2_f3*h))/g1;

        k4_f1 = (Tg1_x[j]*e2+Tg2_x[j]*e3-(Tf1_x[j]+k3_f1*h))/e1;
        k4_f2 = (Tg2_x[j]*f2+Tp_x[j]*f3-(Tf2_x[j]+k3_f2*h))/f1;
        k4_f3 = (Ts*g2+Ta*g3-(Tf3_x[j]+k3_f3*h))/g1;

        Tf1_x[j+1] = Tf1_x[j]+(h/6.0)*(k1_f1+2.0*k2_f1+2.0*k3_f1+k4_f1);
        Tf2_x[j+1] = Tf2_x[j]+(h/6.0)*(k1_f2+2.0*k2_f2+2.0*k3_f2+k4_f2);
        Tf3_x[j+1] = Tf3_x[j]+(h/6.0)*(k1_f3+2.0*k2_f3+2.0*k3_f3+k4_f3);

        Tg2_x[j+1] = ((b1+c1*b2+a1*b4)+Tf1_x[j]*(b3+a3*b4)+Tf2_x[j]*(c3*b2+b5)+Ts*
        Tf1_x[j+1] = a1+Tg2_x[j]*a2+Tf1_x[j]*a3+Ta*a4+Tgkv*a5;
    }
}

```

Figure 5: Runge-Kutta solver in Java code.

code programmed here will be executed once and always before the simulation starts. A snapshot of the model variables initialization can be seen in Figure 3.

The next step is to implement the evolution of the simulation. Mainly, the model is defined using ordinary differential equations (ODEs), and EJS provides a tab to declare the equations that will be automatically solved. Programmer is allowed to choose among different numerical methods to integrate the equations, choosing also the step of integration. Among those methods programmer can find the simple Euler method, Euler-Richardson, Runge-Kutta of different orders, Bogacki-Shampine 3(2), Cash-Karp 5(4), different Fehlberg methods, different Dormand-Prince methods, Radau 5(4), and even a sophisticated QSS 3 method. The editor accepts as many equations are required in the model.

If the model incorporates partial equations (with more than one independent variable), or some ODEs with different independent variable or even if the model has algebraic equations, then the model can be implemented without the editor but using another tab in the same evolution step where Java code is accepted. Also, it is possible to combine both, the Java code and the ODE editor. This is the case in the proposed model in this paper. The ODE editor is used to program the equations in the time as independent variable, but for those equations in the model that use  $x$  as independent variable a numerical method has been programmed.

Figure 4 shows the ODE editor; note the solver method popup menu. Its simplicity makes easier to implement any ODE equations system with an independent variable. In Figure 5 the evolution of the simulation can be programmed directly in Java code; in this case, the figure shows the programming of a Runge-Kutta solver method [32, 33]. The combination

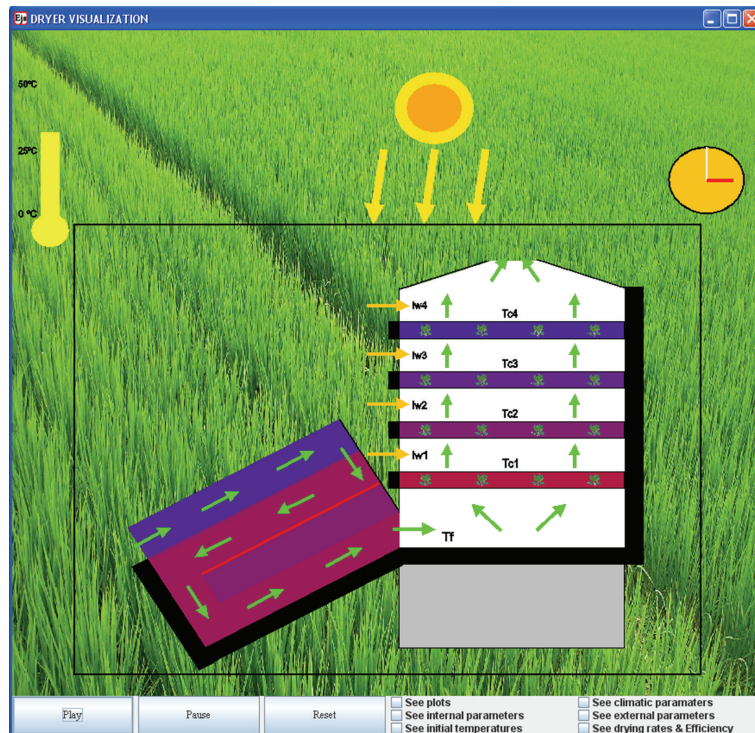


Figure 6: Main view of the solar dryer simulator (SDSim).

of both makes EJS very useful to program models in a modular, coherent programming scheme.

Tabs “Custom” and “Fixed Relations” allow the programmer to insert additional code for functions called from the “Evolution” tab. This code will be programmed in Java.

### 3.2.2. The View

To define the view of the simulation, there is a special editor that allows inserting the graphical elements in a tree structure. Those elements are plots, pictures, and interactive objects.

For the main view, authors have designed a draw with the most important elements in the simulation in the solar drier: the air flow, the multitray, the sun movement, the temperature evolution, and the time. This view is shown in Figure 6. Together with the drawing elements, there are buttons to start, pause, and reset the simulation. As the simulation evolves in time, the different elements associated to the model variables also change their shape or position in the screen. For instance, the sun moves from left to right simulating the day (from east to west), incrementing its size until midday, and decrementing it during the afternoon. Also the arrows, simulating the sun rays, point always to the solar drier as the sun moves. The effect is very attractive and very comprehensive for those who are not used on programming. In this view, it is possible to select among different plots to show that the evolution of the variables respects the independent variable (time  $t$  or position  $x$ ) as

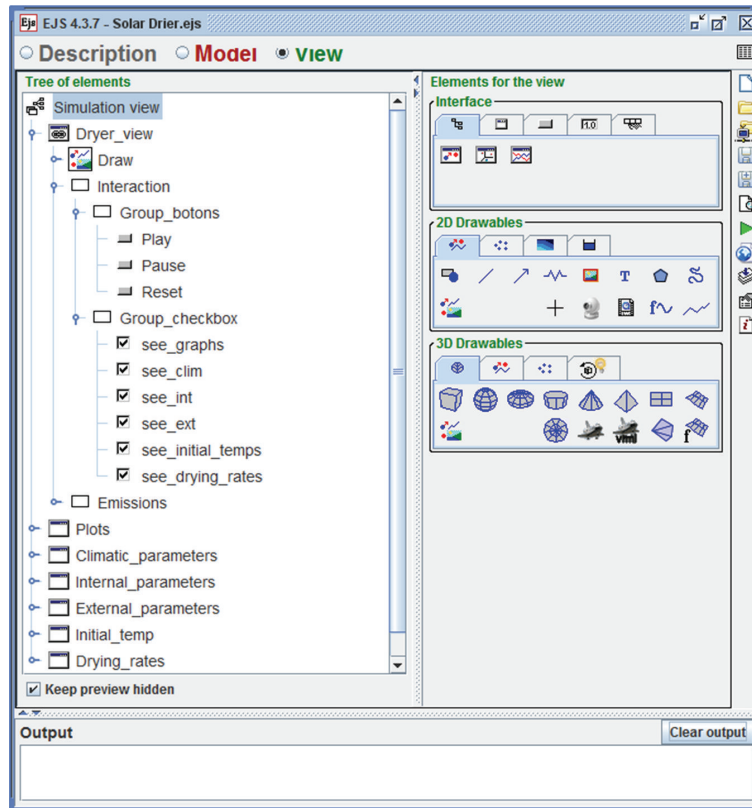


Figure 7: Main menu to create a view.

a classical plot. Moreover, it is possible to open some dialogs to modify the model parameters to change the simulation setup and evolution.

To interface the simulation with the user, six different frames are proposed: the graphical evolution (Drier view); four frames to modify the model parameters (internal parameters, external parameters, climatic parameters, and initial temperatures) and one frame to show numerical plots of output model variables.

To create a view, a tree structure must be followed. In this structure, programmer can simultaneously insert interfaces, 2D drawables, and 3D drawables, depending on the required element and choosing among an already defined library of objects. Figure 7 shows the tree structure used to represent the view of the solar drier and just a few of the elements needed to create the view, plots and dialogs to interface with the user.

Among several elements to show in the simulation evolution, one of the most useful are plots that represent numerically the variables evolution along the simulation-independent variable. Firstly, a Plotting Panel has to be inserted in the tree of elements in a desired position. Then a trace element has to be inserted within the plotting panel indicating which variable must be represented. In Figure 8, the trace of variable  $T_a$  respect the time has been inserted in a plotting panel.

To draw the main view, the solar drier, a 2D-drawable element is required, and specifically is the element "shape," see Figure 9. In this element, it is possible to indicate the position, size, scale, and different transformations for a predefined shape. In this case

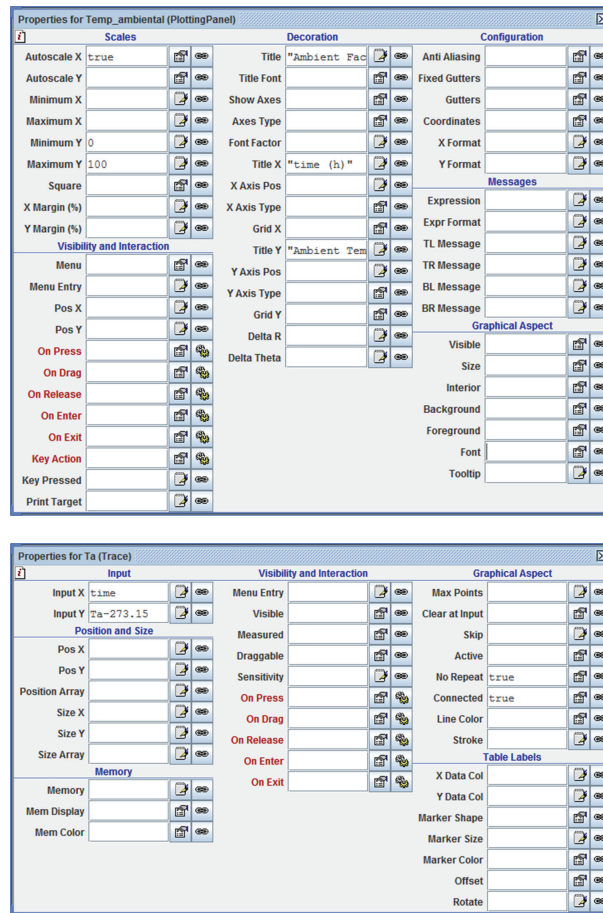


Figure 8: EJS elements to create a plotting panel and a trace within it.

the shape is a rectangle filled in with a color indicated in the variable “color\_g1” that changes along the simulation evolution. The remainder elements in the main view are built in a similar way (round shape for the sun and the clock, arrow elements for the sun rays, rectangle for the temperature...) with varying parameters associated to simulation variables. The final view is quite laborious because all the elements must coincide and evolve properly, but the result is sometime astonishing for the user.

#### 4. Simulation Results

Before executing the simulation, user has to initialize the different parameters and variables involved in the drying process. More specifically, climatic conditions, external parameters, internal parameters, and the initial values of the temperatures need to be set. Once all these parameters have been selected, the user can initiate the simulation by clicking on the play button.

Among the different variables studied in the simulation, it is necessary to differentiate between those variables in which evolution depends only on the time  $f(t)$  and those which



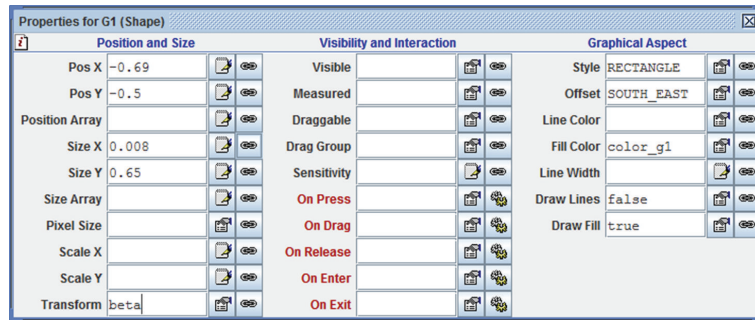


Figure 9: Element to represent shapes.

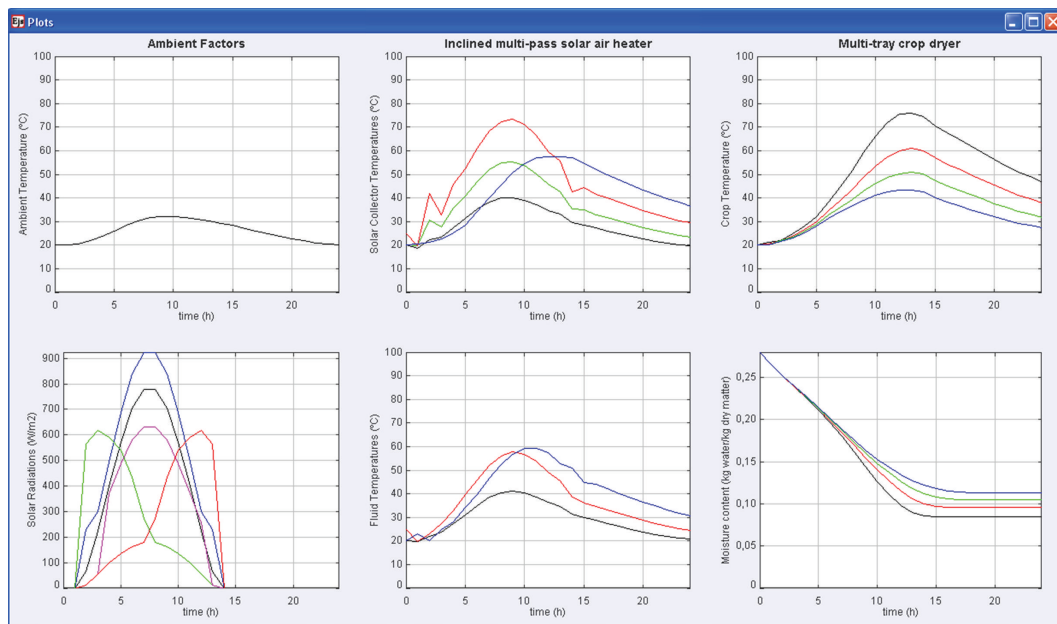


Figure 10: Plots representing output model variables.

evolve also as a function of position  $f(x, t)$ . Except the storage material, which is considered to be at the same temperature in each instant of time, the rest of temperatures of the multipass air heater depend on both time and position. In contrast, the temperatures of the crops and their moistures are only a function of time.

Easy Java does not permit to solve partial differential equations or ordinary differential equations with different independent variable using the ODE editor, and only it is allowed to solve ordinary differential equations with one independent variable. In our case, to solve the equations in time  $t$ , the ODE editor is used with Fehlberg-8(7) numerical method. Instead, when the equations to be solved are in the independent variable  $x$ , a 4th order Runge-Kutta is implemented in Java, see Figure 5. Both methods give accurate results and satisfactory solutions [32, 33].

The simulation begins ( $t = 1$  hour) with the computation of the time-dependent variables which are the storage material temperature ( $T_s$ ), the crop temperatures

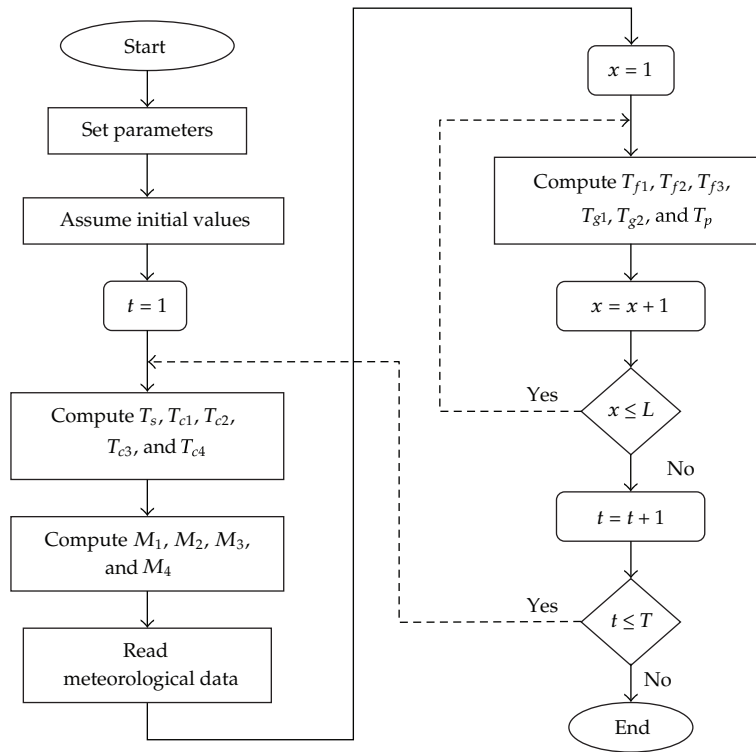


Figure 11: Simulation flux diagram.

$(T_{c1}, T_{c2}, T_{c3},$  and  $T_{c4})$ , and the crops moisture content ( $M_1, M_2, M_3,$  and  $M_4$ ). Once they are calculated, the external meteorological data (solar radiation and ambient temperature) is read. After this, the position is initialized ( $x = 1$ ), and the rest of the collector temperatures are calculated ( $T_{g1}, T_{g2}, T_p, T_{f1}, T_{f2},$  and  $T_{f3}$ ). Then, the time remains fixed but the position  $x$  is incremented, evaluating these collector temperatures for different positions of  $x$  until the total length  $L$  is reached, the integration step for  $x$  can be chosen by the programmer. Finally, the time is incremented, and this process is repeated for each hour up to the maximum time established  $T$ , which is a complete day (24 h) for the images shown in this paper. Figure 11 represents the flux diagram of execution.

It has to be noticed that the increment of time and the computation of the time depending variables is done in the same step by the ODE solver of the Easy Java Simulation.

The main view of the simulation can be seen in Figure 6. The view represents the solar drier and all the changing elements during the simulation: air flow in the heater, temperatures in the heater and the drier, the sun size and position, the sun rays, the clock, the thermometer, and air flow along the trays. Each graphical element is associated to an output variable, and then the elements move and evolve in the view. At the bottom of the main view, there are the starting, pausing, and resetting buttons as well as the boxes to select the windows to see the plots or the dialogs. If plots are selected then the plots in Figure 10 pop in the screen showing six plots representing the most relevant variables in the model. The plot in the upper right position shows the evolution of the temperature in each tray where the crops are located to be dried. If a click is done on the plot, it pops a more detailed image, shown in Figure 12. In

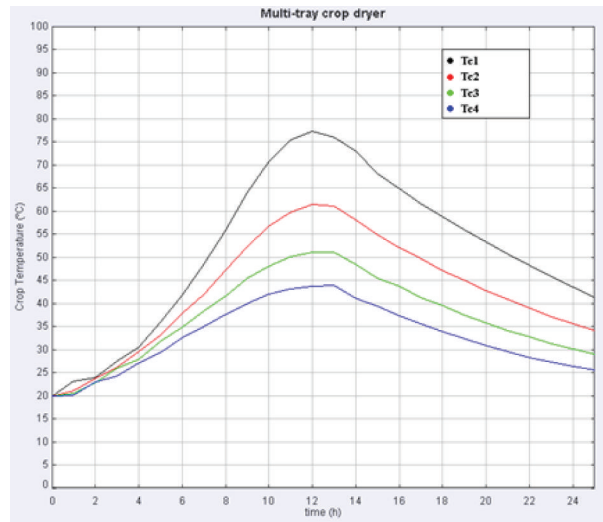


Figure 12: Enlargement of the multitray crop dryer temperature.

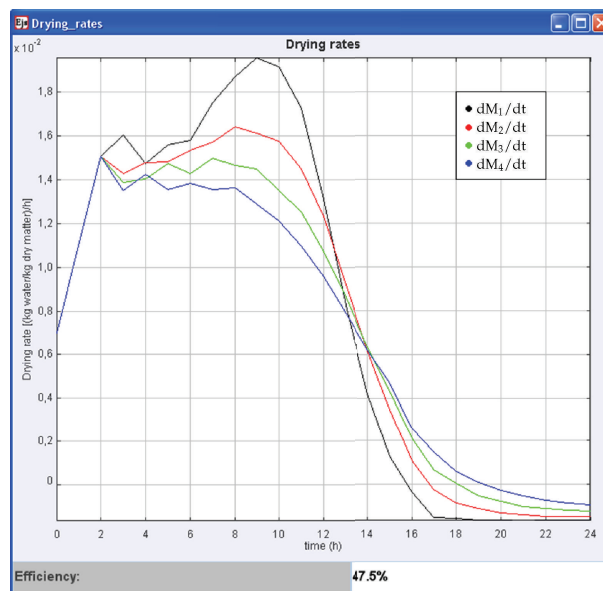


Figure 13: Variation of the moisture contents.

this new image, the legend for each trace appears in the plotting area allowing knowing and following the temperature for tray along the simulation time.

A very interesting result of the simulation is the moisture contents of the different crops placed in the trays. The evolution of this variable,  $M$ , can be seen in Figure 10 lower right, and it indicated how the moisture contents decrease with time until a final value that it is considered the final moisture for that crops. From this bottom value, the moisture reaches the equilibrium moisture contents,  $M_e$  in the model, that it is a value that depends on the external factors such as the temperature. In this example, for a rough rice crop, the

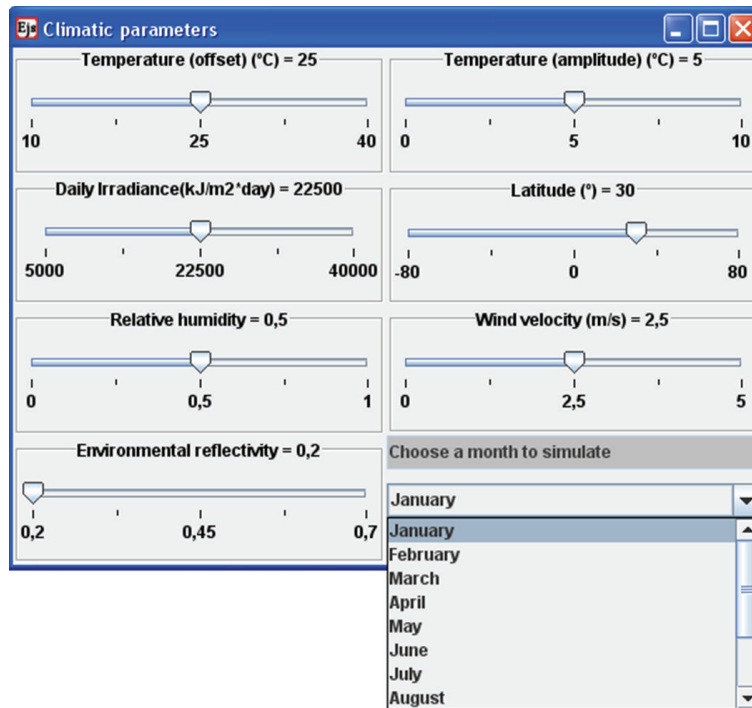


Figure 14: Dialog to modify the climatic parameters.

equilibrium varies between 0.08 and 0.10 kg water/kg of dry matter in the lower tray (the most effective in the drying process) [9, 26].

It is also of interest to plot the variation of the moisture contents ( $dM/dt$ ) along the time to see the most effective hours of drying. As it was expected, the solar drier accumulates heat during the day and reaches its maximum drying effect in the afternoon, with a peak at 3.00 PM. The plot can be seen in Figure 13 for each tray in the drier. At the bottom of the figure, the efficiency is displayed. This value is calculated following the expression for  $\eta_0$  in the model equations. The interactive simulator allows the engineers and drier designers to change the parameters in order to find the most efficient design for new solar driers. This is the final goal of the simulator.

All the results and values for the output variables have been validated with the real results in [19], showing that the model behaves as it is expected corresponding to the collected experimental data in the solar drier. If the model works properly with the real data and the validation is correct, then the parameters can be changed and the new results are expected to correspond to the design of the resulting solar drier with those modified parameters.

One important feature of EJS is the interactivity that user can have with the simulation, and it is achieved with the modification of the model parameters during the simulation. In Figures 14, 15, and 16, the current values of different model parameters are displayed and, using the sliders, the pop-up menus or the numerical fields, they can be changed. Once they are changed, the simulation resumes executing with the modified values, changing the evolution according to these new parameter values.

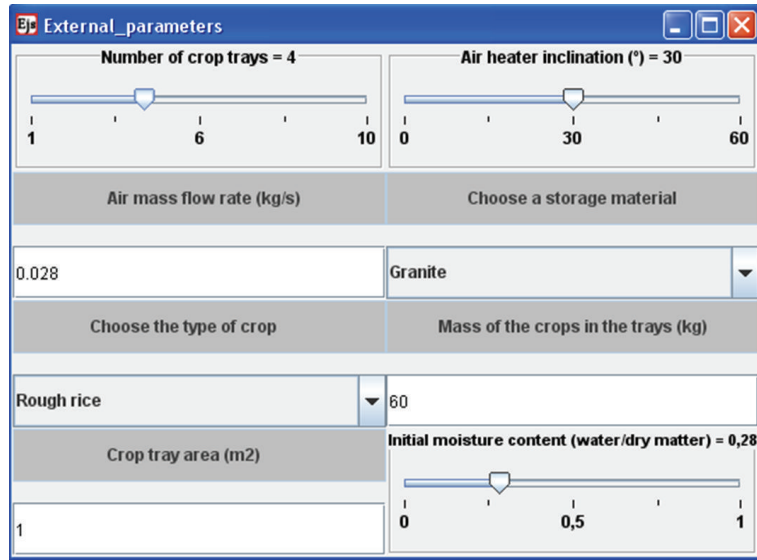


Figure 15: Dialog to modify the external parameters.

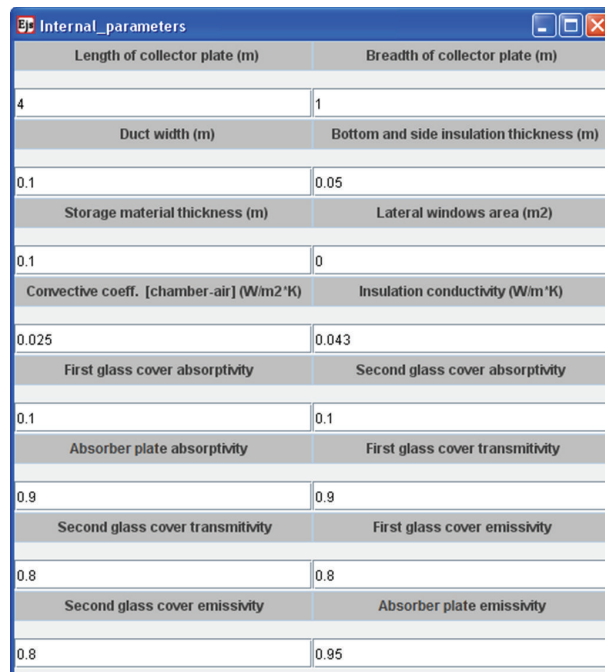


Figure 16: Dialog to modify the internal parameters.

In order to set the input variables to the model (temperature and solar radiation), the user can choose between to use a mathematical function to generate those values or to choose a database with experimental data [28, 29, 34]. This aspect enriches the simulator, and it is very useful to test real designs with actual collected data.

## 5. Conclusions

In this research paper, authors have implemented a novel kind of simulator or solar driers. The model is based in a real drier in order to validate the model and the simulator. This model is based in a multicrop, inclined multipass solar air heater with in-built thermal storage drier. The simulator has the ability to interact with the user, and the model parameters can be changed during the evolution time modifying the simulation results. The objective of the simulator is to find the most efficient design for new solar driers.

## Nomenclature

$A$ :	Area, $m^2$
$A, B, D, b'$ :	Constants
$a_w$ :	Water activity, decimal
$b$ :	Breadth of collector plate, m
$C$ :	Specific heat at constant pressure, $J \cdot kg^{-1} \cdot K^{-1}$
$d$ :	Duct width, m
$g$ :	Acceleration due to gravity, $m \cdot s^{-2}$
$hc$ :	Convective heat transfer coefficient, $W \cdot m^{-2} \cdot K^{-1}$
$h_c$ :	Convective heat and mass transfer coefficient from crop to air, $W \cdot m^{-2} \cdot K^{-1}$
$h_{ch}$ :	Convective heat loss coefficient from chamber to air, $W \cdot m^{-2} \cdot K^{-1}$
$hr$ :	Radiative heat transfer coefficient, $W \cdot m^{-2} \cdot K^{-1}$
$h_o$ :	Convective and radiative heat transfer coefficient due to wind, $W \cdot m^{-2} \cdot K^{-1}$
$I_h$ :	Hourly average solar radiation on horizontal surface, $W \cdot m^{-2}$
$I_t$ :	Hourly average solar radiation on tilted surface, $W \cdot m^{-2}$
$I_v$ :	Hourly average solar radiation on vertical surface, $W \cdot m^{-2}$
$K$ :	Conductivity, $W \cdot m^{-1} \cdot K^{-1}$
$K_d$ :	Drying constant, $s^{-1}$
$L$ :	Length of collector plate, m
$L_c$ :	Latent heat of vaporization of moisture from crop, $J \cdot kg^{-1}$
$l$ :	Thickness, m
$M$ :	Moisture content of grain, kg water/kg of dry matter
$M_c$ :	Mass of the crop in the trays, kg
$M_e$ :	Equivalent moisture content, kg water/kg of dry matter
$M_{ev}$ :	Moisture evaporation per hour, $kg \cdot water \cdot h^{-1}$
$m_s$ :	Mass of storage material, kg
$m_a$ :	Mass flow rate, $kg \cdot s^{-1}$
$Nu$ :	Nusselt number
$Q_u$ :	Rate of useful thermal energy from solar air heater, W
$Ra$ :	Rayleigh number
$T$ :	Temperature, K
$T_c$ :	Temperature, $^{\circ}C$
$\Delta T$ :	Temperature difference, K
$t$ :	Time, s
$U_b$ :	Bottom or side loss coefficient, $W \cdot m^{-2} \cdot K^{-1}$
$U_i$ :	Overall heat loss coefficient from the glass windows, $W \cdot m^{-2} \cdot K^{-1}$
$v$ :	Wind velocity, $m \cdot s^{-1}$
$x$ :	Length of coordinate in direction of flow, m.

*Greeks*

$a$ :	Absorptivity
$\alpha_f$ :	Diffusivity of air, $\text{m}^2 \text{s}^{-1}$
$\beta$ :	Tilt angle of collector, degree
$\beta'$ :	Expansion factor, $\text{K}^{-1}$
$\varepsilon$ :	Emissivity
$\gamma$ :	Relative humidity of air, decimal
$\eta_o$ :	Overall thermal efficiency, %
$\nu_f$ :	Kinematic viscosity of air, $\text{m}^2 \text{s}^{-1}$
$\rho$ :	Density, $\text{kg}\cdot\text{m}^{-3}$
$\sigma$ :	Stefan-Boltzmann constant, $\text{W}\cdot\text{m}^{-2}\cdot\text{K}^{-4}$
$\tau$ :	Transmissivity.

*Subscripts*

$a$ :	Air
$b$ :	Bottom and side insulation of collector and dryer
$ba$ :	Bottom of insulation to air
$c$ :	Crop
$c1, c2, c3, c4$ :	Crop in tray 1, 2, 3, 4
$ch$ :	Chamber
$f$ :	Fluid (air)
$f1$ :	Air stream-I
$f1g2$ :	Air in stream-I to second glass cover
$f2$ :	Air stream-II
$f3$ :	Air stream-III
$g1$ :	First glass cover
$g1a$ :	First glass cover to air
$g1f1$ :	First glass cover to air in stream I
$g1sky$ :	First glass cover to sky
$g2$ :	Second glass cover
$g2g1$ :	Second glass cover to first glass cover
$g2f2$ :	Second glass cover to air in stream II
$i$ :	Sides of glass window, integer 1, 2, 3
$j$ :	Number of trays, integer 1, 2, 3, 4
$p$ :	Absorber plate
$pf2$ :	Absorber plate to air in stream II
$pg2$ :	Absorber plate to second glass cover
$ps$ :	Absorber plate to storage material
$s$ :	Storage material
$sf3$ :	Storage material to air in stream III
$sky$ :	Sky
$w$ :	Window
$w1, w2, w3, w4$ :	Window of tray 1, 2, 3,4.

## Acknowledgments

This work has been partially funded by the Technical University of Catalonia under the Project UPC-ICE "2011 Pedagogical Innovation Projects."

## References

- [1] I. Farkas, "Solar drying," *Stewart Postharvest Review*, vol. 4, no. 2, article 9, 2008.
- [2] K. R. Manohar and P. Chandra, "Drying of agricultural produce in a greenhouse type solar dryer," *International Agricultural Engineering Journal*, vol. 9, no. 3, pp. 139–150, 2000.
- [3] S. Aboul-Enein, A. A. El-Sebaei, M. R. I. Ramadan, and H. G. El-Gohary, "Parametric study of a solar air heater with and without thermal storage for solar drying applications," *Renewable Energy*, vol. 21, no. 3-4, pp. 505–522, 2000.
- [4] M. M. Alkilani, K. Sopian, M. A. Alghoul, M. Sohif, and M. H. Ruslan, "Review of solar air collectors with thermal storage units," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 3, pp. 1476–1490, 2011.
- [5] O. V. Ekechukwu and B. Norton, "Review of solar-energy drying systems II: an overview of solar drying technology," *Energy Conversion and Management*, vol. 40, no. 6, pp. 615–655, 1999.
- [6] O. V. Ekechukwu and B. Norton, "Review of solar-energy drying systems III: Low temperature air-heating solar collectors for crop drying applications," *Energy Conversion and Management*, vol. 40, no. 6, pp. 657–667, 1999.
- [7] I. Farkas, "Integrated use of solar energy crop drying," in *Proceedings of the European Drying Conference (EuroDrying '11)*, Palma de Mallorca, Spain, 2011.
- [8] H. R. Bolin and D. K. Salunkhe, "Food dehydration by solar energy," *Critical reviews in food science and nutrition*, vol. 16, no. 4, pp. 327–354, 1982.
- [9] M. A. Basunia and T. Abe, "Thin-layer solar drying characteristics of rough rice under natural convection," *Journal of Food Engineering*, vol. 47, no. 4, pp. 295–301, 2000.
- [10] T. M. DeJong, "Peach: peach crop yield and tree growth simulation model for research and education," in *Proceedings of the 5th International Symposium on Computer Modelling in Fruit Research and Orchard Management*, Wageningen, The Netherlands, October 1999.
- [11] P. Fellows, "Guidelines for small-scale fruit and vegetable processors," FAO Agricultural Services Bulletin 127, FAO of the United Nations, Rome, Italy, 1997.
- [12] G. Belforte, F. Dabbene, and P. Gay, "LPV approximation of distributed parameter systems in environmental modelling," *Environmental Modelling and Software*, vol. 20, no. 8, pp. 1063–1070, 2005.
- [13] G. S. McMaster, J. C. Ascough, D. A. Edmunds, A. A. Andales, L. Wagner, and F. Fox, "Multi-crop plant growth modeling for agricultural models and decision support systems," in *Proceedings of the International Congress on Modeling and Simulation Proceedings*, December 2005.
- [14] CropSyst, [http://www.bsyse.wsu.edu/cs\\_suite/cropsyst/](http://www.bsyse.wsu.edu/cs_suite/cropsyst/).
- [15] C. O. Stöckle, M. Donatelli, and R. Nelson, "CropSyst, a cropping systems simulation model," *European Journal of Agronomy*, vol. 18, no. 3-4, pp. 289–307, 2003.
- [16] M. Condori and L. Saravia, "The performance of forced convection greenhouse driers," *Renewable Energy*, vol. 13, no. 4, pp. 453–469, 1998.
- [17] R. K. Goyal and G. N. Tiwari, "Parametric study of a reverse flat plate absorber cabinet dryer: a new concept," *Solar Energy*, vol. 60, no. 1, pp. 41–48, 1997.
- [18] D. Jain and G. N. Tiwari, "Thermal aspects of open sun drying of various crops," *Energy*, vol. 28, no. 1, pp. 37–54, 2003.
- [19] D. Jain and R. K. Jain, "Performance evaluation of an inclined multi-pass solar air heater with in-built thermal storage on deep-bed drying application," *Journal of Food Engineering*, vol. 65, no. 4, pp. 497–509, 2004.
- [20] D. Jain, "Modeling the system performance of multi-tray crop drying using an inclined multi-pass solar air heater with in-built thermal storage," *Journal of Food Engineering*, vol. 71, no. 1, pp. 44–54, 2005.
- [21] C. Ratti and A. S. Mujumdar, "Solar drying of foods: Modeling and numerical simulation," *Solar Energy*, vol. 60, no. 3-4, pp. 151–157, 1997.



- [22] M. S. Sodha, A. Dang, P. K. Bansal, and S. B. Sharman, "An analytical and experimental study of open sun drying and a cabinet tyre drier," *Energy Conversion and Management*, vol. 25, no. 3, pp. 263–271, 1985.
- [23] O. Yaldiz, C. Ertekin, and H. I. Uzun, "Mathematical modeling of thin layer solar drying of sultana grapes," *Energy*, vol. 26, no. 5, pp. 457–465, 2001.
- [24] A. Whiller, *Design Factors Influencing Solar Collectors, Low Temperature Engineering Application of Solar Energy*, American Society of Heating, Refrigerating and Airconditioning Engineers, New York, NY, USA, 1967.
- [25] J. A. Duffie and W. A. Beckman, *Solar Engineering and Thermal Process*, John Wiley and Son, New York, NY, USA, 2nd edition, 1991.
- [26] A. Iguaz and P. Virseda, "Moisture desorption isotherms of rough rice at high temperatures," *Journal of Food Engineering*, vol. 79, no. 3, pp. 794–802, 2007.
- [27] S. M. Henderson, "A basic concept of equilibrium moisture," *Agricultural Engineering*, vol. 33, pp. 29–32, 1952.
- [28] B. H. Khan, *Non Conventional Energy Sources*, Tata Mc-Graw Hills, 2nd edition, 2009.
- [29] N. Z. Al-Rawahi, Y. H. Zurigat, and N. A. Al-Azri, "Prediction of hourly solar radiation on horizontal and inclined surfaces for Muscat/Oman," *The Journal of Engineering Research*, vol. 8, no. 2, pp. 19–31, 2011.
- [30] F. Esquembre, "Easy Java Simulations: a software tool to create scientific simulations in Java," *Computer Physics Communications*, vol. 156, no. 2, pp. 199–204, 2004.
- [31] Easy Java Simulations, EJS, <http://fem.um.es/Ejs/>.
- [32] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, vol. 8, Springer, Berlin, Germany, 2nd edition, 1993.
- [33] R. K. Goyal and G. N. Tiwari, "Effect of thermal storage on the performance of a deep bed drying system," *International Journal of Ambient Energy*, vol. 20, no. 3, pp. 125–136, 1999.
- [34] J. D. Aguilar, P. J. Perez, J. De la Casa, and C. Rus, "Cálculo de la energía generado por un sistema fotovoltaico conectado a red: Aplicación docente," Universidad de Jaén, 2006.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

