Solar Cooker Project

Anna Buchele, Serena Chen, Yoonyoung Cho

October 28 2015

1 Overview

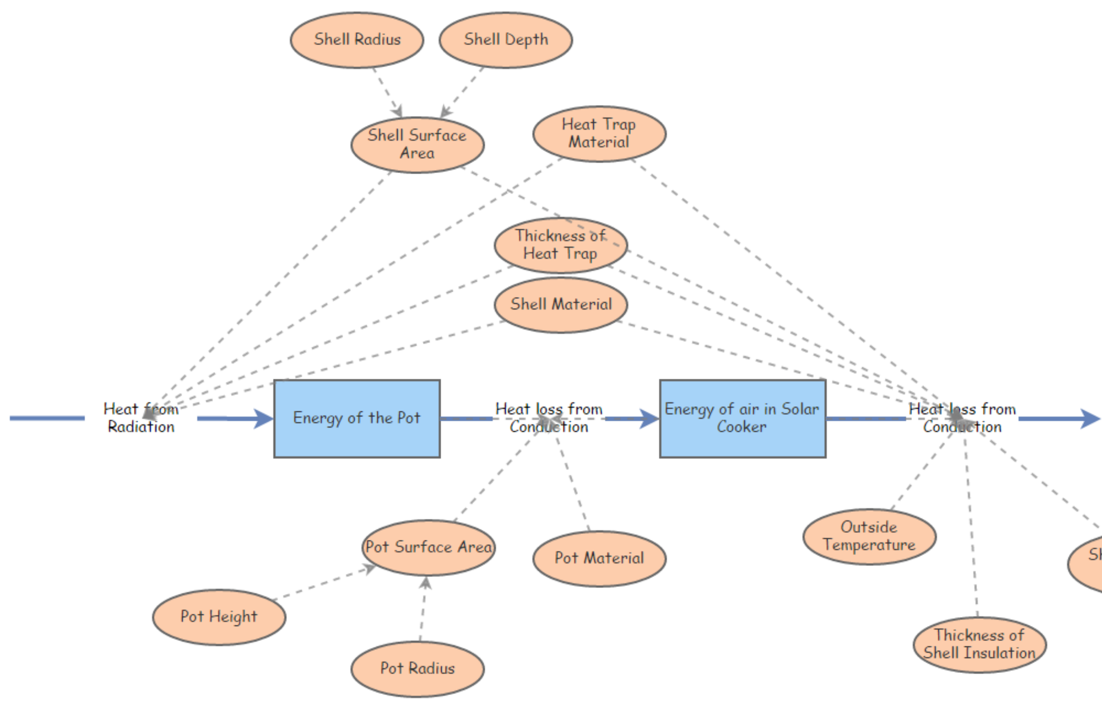

Figure 1: Stock and Flow Diagram of the solar cooker model.

The objective of our design was to achieve sufficient performance (rapid cooking and heat retention) while balancing its cost. In our model, the control parameter was the dimensions of the Solar-cooker insulation other constants are either designated in accordance to material properties, or are assigned empirical values within reasonable range. The success of a model will be gauged by the product of performance and attributes: while performance will be taken as the inverse of the time it took to reach the cooking temperature, as for attributes – which may include price, durability, weight, etc., – the methodology of quantizing is yet to be determined.

2 Relationships

$$A_{par} = \frac{\pi r}{6h^2}((r^2 + 4h^2)^{3/2} - r^3) \tag{1}$$

$$Q_{cond} = \frac{kA\Delta T}{d} \tag{2}$$

$$Q_{rad} = \pi r_{shell}^2 * \frac{k_i}{m^2} * sin(\theta) \tag{3}$$

$$\frac{dT_{pot}}{dt} = Q_{rad} - Q_{cond_{p,a}} \tag{4}$$

$$\frac{dT_{air}}{dt} = Q_{cond_{p,a}} - Q_{cond_{a,e}} \tag{5}$$

Equation (1) describes the relationship of the surface area of the paraboloid(the solar cooker) in terms of depth and the radius; (2) is a standard heat-flow equation for conduction. (3) describes the absorption of heat from insolation, while (4) and (5) encapsule the above equations into a singular flow for the respective stocks.

3 Constants

3.1 Absolute values

$$k_i = 200W/m^2$$
$$T_e = 281K$$
$$c_{air} = 1005J/kgK$$
$$\rho_{air} = 1.205kg/m^3$$

Here, the insolation value(k_i) and the environment temperature(T_e) was determined by an annual average[1]; the specific heat of air(c_{air}) is relatively constant, as is its density(p_{air}), regardless of location.

3.2 Assigned values

$$r_s = .48m$$
$$h_s = 0.28m$$
$$r_p = 0.1m$$
$$h_p = 0.2m$$
$$d_p = 0.03m$$

In the above representation, r,h,and d denotes radius, height, and thickness, respectively; s and p indicate shell(outer surface of the solar cooker) and pot(inner bowl). Here, the radius and height refer to an existing Cooker[2]; the pot's dimensions– the size and thickness – were chosen to represent a standard. In order to visualize whether or not the dimensions were practical, the CAD model of corresponding dimensions was created:

[1] http://www.hindawi.com/journals/isrn/2012/203149/
[2] http://sunoven.com/sun-cooking-usa/how-to-use/faq/

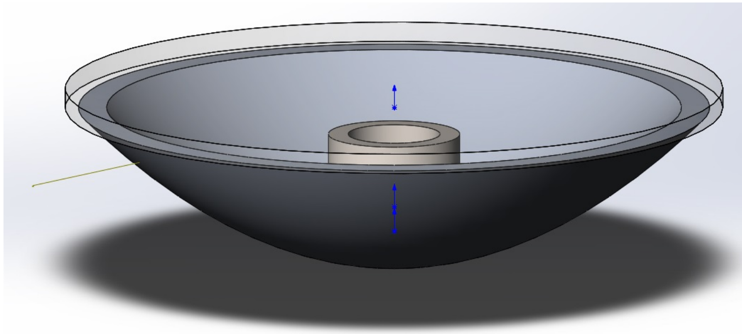

Figure 2: The CAD model of the Solar Cooker.

3.3 Material Properties

Material	Thermal Conductivity (W/mK)	Emmissivity	$/lbs
Stainless Steel	16	0.85	0.25
Cast Iron	55	0.60-0.70	0.70
Copper	401	0.78	2.00
Aluminium	205	0.039 - 0.057	0.40
Copper-Bronze	110	0.10	0.50
Copper-Brass	109	0.22	1.10
Porcelain	1.5	0.93	-

Table 1: A Selection of Material Properties

Of the above values, only the Cast Iron was thoroughly simulated hereto. Although the primary concern of our research is the thickness of insulation, further research may be conducted as to the most appropriate insulation material as well.

4 Implementation

After determining the constants, the model was transcribed in MATLAB for simulation.

`paramInit.m` :

```
% Script to set parameters

% cooker dimensions
% taken from http://sunoven.com/sun-cooking-usa/how-to-use/faq/
params.radius = 2*19*2.54/100; %meters
params.depth = 2*11*2.54/100; %meters

% pot dimensions
params.potRad = .1; %meters
params.potHeight = .2; %meters
params.potWallWidth = .03; %meters


% material of the pot: Cast Iron
% k = 27-46
% e = .6-.7
% density = 6800 - 7800
```

```
% c = 500
params.kPot = 35*1.731; %W/(mK)
params.efficAbsorbtion = .65; %constant
params.potDensity = 7300; %kg/m^3
params.potSpecHeat = 500; %J/KgK

% air constants
params.insolation = 200; %W/m^2
params.airSpecHeat = 1005; %J/KgK
params.airDensity = 1.205; %kg/m^3

% Average 47 degress F
params.tempOutside = 281.483; %K

params.endTime = 7200;
params.startTemp = params.tempOutside;
```

solarCookerTimeSeries.m :

```
function solarCookerTimeSeries(k, d)
    paramInit;
    %initialize thermal conductivity for the material
    %initialize thickness of walls
    params.thermConductCooker = k;
    params.width = d;

    mcPot = potSurfArea(params.potRad, params.potHeight)*params.potWallWidth*params.potDensity*params.potSpecHeat;
    mcAir = (cookerVol(params.radius, params.depth)-potVol(params.potRad, params.potHeight))*params.airDensity*params.airSpecHeat;

    [T1,Y1] = ode45(@flows, [0, params.endTime], [T2U(params.startTemp, mcPot) T2U(params.startTemp, mcAir)]);
    plot(T1, U2T(Y1(:,1), mcPot));
    hold on
    plot(T1, U2T(Y1(:,2), mcAir));
    xlabel('Time');
    ylabel('Temperature');
    legend('pot','inside air');
    title('Solar Cooker Temperature Time Series');
%% Flow function

% y is a vector containing [energy of the pot, energy of the air]
function res = flows(t, y)
    potEnergy = y(1);
    airEnergy = y(2);

    tempPot = U2T(potEnergy, mcPot);
    tempAir = U2T(airEnergy, mcAir);
    %dpot/st = radiation-conduction
    %assumption: sun is straight up and down. Also no shadows. Also volume
    %of the pot = surface area * thickness

    radiation1 = radiation(params.efficAbsorbtion, params.insolation, pi*params.radius^2);
```

```
    conduction1 = conduction(params.kPot, potSurfArea(params.potRad, params.potHeight), params.potWallWidth, tempPot-tempAir);
    conduction2 = conduction(params.thermConductCooker, cookerSurfArea(params.radius, params.depth), params.width, tempAir-params.tempOutside);

    %[tempAir U2T(conduction2, mcPot)]
    res(1) = radiation1 - conduction1;
    res(2) = conduction1 - conduction2;
%     msg = sprintf('%d %d %d %d', tempPot, tempAir, res(1)/mcPot, res(2)/mcAir);
%     disp(msg);
%     res(1) = radiation(params.efficAbsorbtion, params.insolation, pi*params.radius^2) + ...
%         conduction(params.kPot, potSurfArea(params.potRad, params.potHeight), params.potWallWidth, tempPot-tempAir);
%     res(2) = conduction(params.kPot, potSurfArea(params.potRad, params.potHeight), params.potWallWidth, tempAir-tempPot) + ...
%         ;
    res = res';
end
%% Formula for energy/temp conversion

% energy to temp
function res = U2T(energy, mc)
    res = energy./mc;
end

% temp to energy
function res = T2U(temp, mc)
    res = temp.*mc;
end
%% Basic formulas to find area/volume

% Solves for the volume of the solar cooker, given the radius and height,
% and assuming a paraboloid shape
function res = cookerVol(rad, dep)
    res = (pi/2)*dep*rad^2;
end

% Solves for the surface area of the solar cooker, given the radius and
% height, and assuming a paraboloid shape.
function res = cookerSurfArea(rad, dep)
    res = (pi*rad/(6*dep^2))*((rad^2 + 4*dep^2)^(3/2) - rad^3);
end

function res = potSurfArea(rad, h)
    res = 2*pi*rad^2 + 2*pi*rad*h;
end

function res = potVol(rad, h)
    res = h*pi*rad^2;
end

%% Basic formulas for heat transfer
```

```
% tempDiff = material - outside
function res = conduction(thermConduct, surArea, wallWidth, tempDiff)
    res = thermConduct*surArea*tempDiff/wallWidth;
end

% tempDiff is air - material
function res = convection(heatTransCoef, surArea, tempDiff)
    res = heatTransCoef*surArea*tempDiff;
end

function res = radiation(efficAbsorb, insolation, area)
    res = efficAbsorb*insolation*area;
end

end
```

In this particular simulation, the thickness was set to be 0.03m, and the conductivity 0.1 W/mK. The resultant plot:

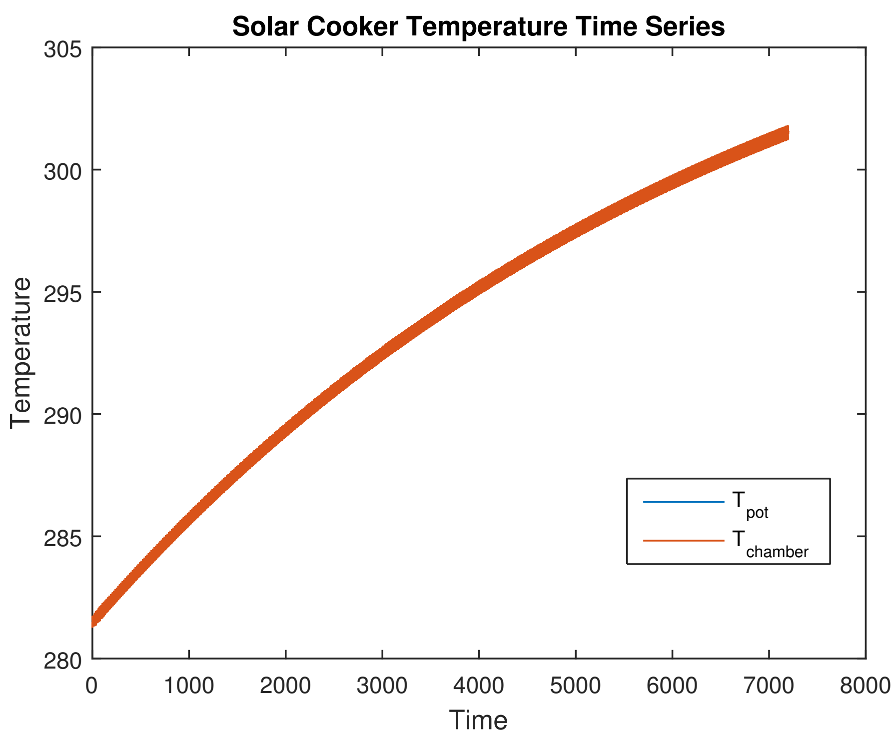

Figure 3: The temperature of the system over time. The plot is shown in a thick line, in which the plot for the pot is embedded, due to the rapid fluctuation of air temperature in the chamber.

This result is far from ideal: the pot fails to reach boiling temperature for 12 hours. In further iterations of the model, the parameters will be explored to find a practical solution that balances its performance with the expenditure – be it weight, durability, or incurred cost.