ENER-GUYZ CONSULTATION

In Cooperation With Novascular_Productions
4/6/2021

By: Nathan Delos Santos
Section 6

What is happening here is given the rate of power produced, I need to find the total amount of power produced in an hour. Integrating the rate tells me how much power is produced at each instant. For example, at t = 0, I might be producing 1kW/min. But at that exact instant, I might have produced 0.5kW. This is why to find the total amount of power produced, I need to integrate twice. I need to add up all the power produced at every instance.

A simpler example:
You are trying to find position traveled given acceleration. You know that the gravity, which increases the velocity at a constant rate, is 9.81m/s^2 (We define down as positive here). Without using kinematics equations, find the total distance traveled from 0 to 10 seconds.

Acceleration:

$$\frac{dy}{dt} = 9.81$$

Now that you know the rate at which velocity changes, you now know the velocity at each point. Now that you know the speed at each point in time, you can then calculate the total distance traveled. Integrating acceleration to get the velocity, you get:

Velocity:

$$v = 9.81t$$

Now that you know how fast it travels each interval, you can add those values up to find the total distance traveled.You end up with

Position:

$$y = \frac{9.81}{2}t$$

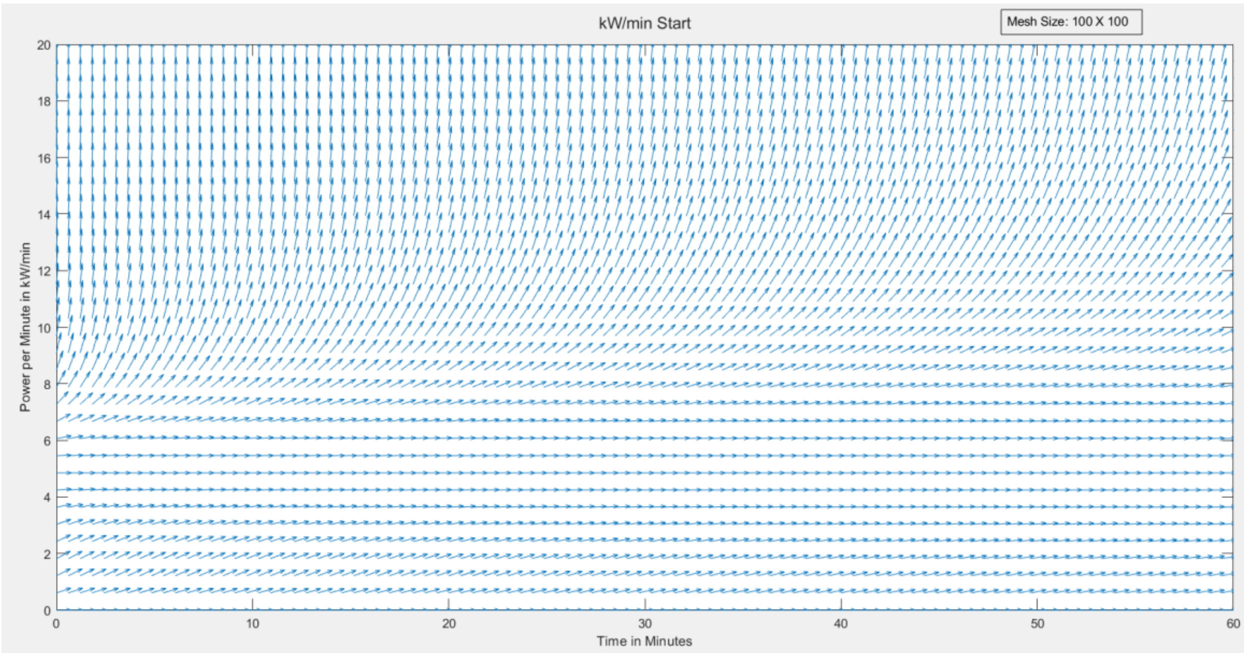
Plug in the bounds to know how far you traveled. You end up traveling 49.05m

Total Distance Traveled:

$$y = \frac{9.81}{2}(10) - \frac{9.81}{2}(0)$$

$$y = 49.05$$

Unfortunately, I was not able to solve the DE by hand, so I used an approximate solution. I used what is known in Calculus as a “slope field”. This is a graph, in which at each point, an arrow points in the direction a curve passing through it will be tangent to. In simpler terms, it expresses the rate of change at each point, and restricts curves to be tangent to them.



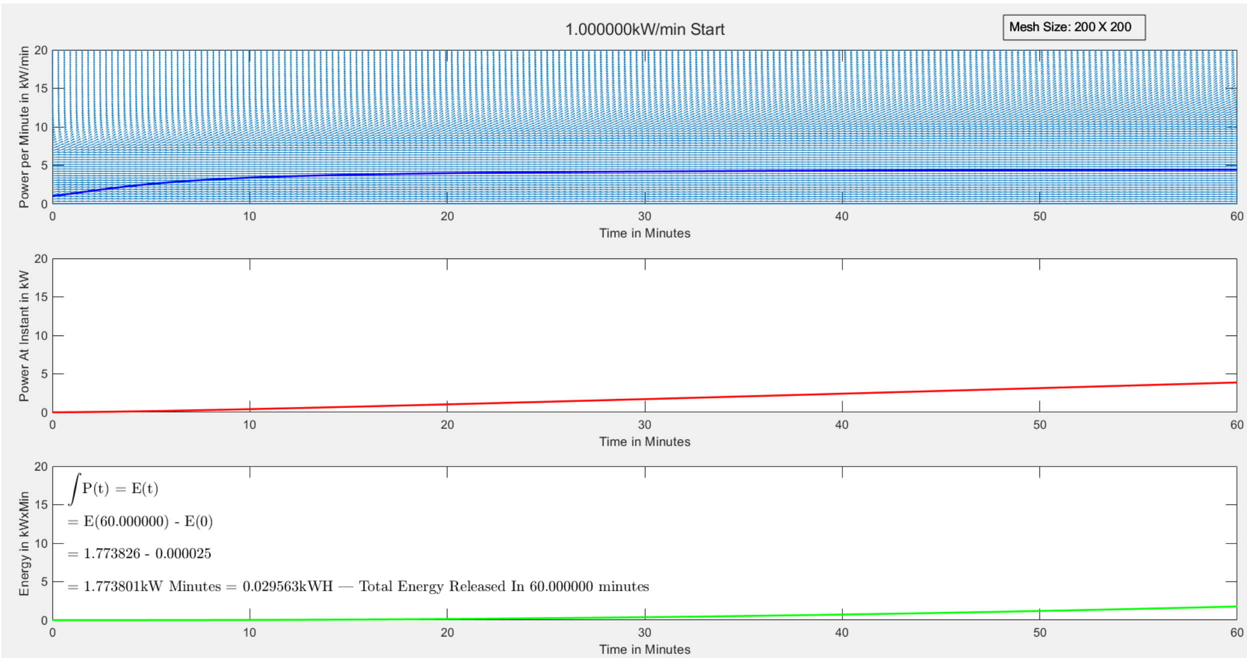
With this visual, it is clear to see how lines would be confined to the arrows. There are an infinite number of curves that can pass through this slope field. This is clearly seen with the arrows at each point on the graph. However, to get exactly one solution out of it, the curve must have a start condition. This is what the y variable, in this case, P, is when x, or in this case, t, is 0.

To make up for this approximate solution, I used many increments to get a finer and finer mesh every time, eventually, settling on a 200 x 200 mesh. This mesh divided 60 minutes into 200 evenly spaced intervals, giving me an accurate plot for all three graphs of: rate, instant, and total(energy). I decided not to go past that size of mesh, because the results beyond it were similar enough to when the mesh was 200 x 200, and took up more computing power. In addition, it made the vectors on the slope field indistinguishable from each other, because they all crowded each others’ space

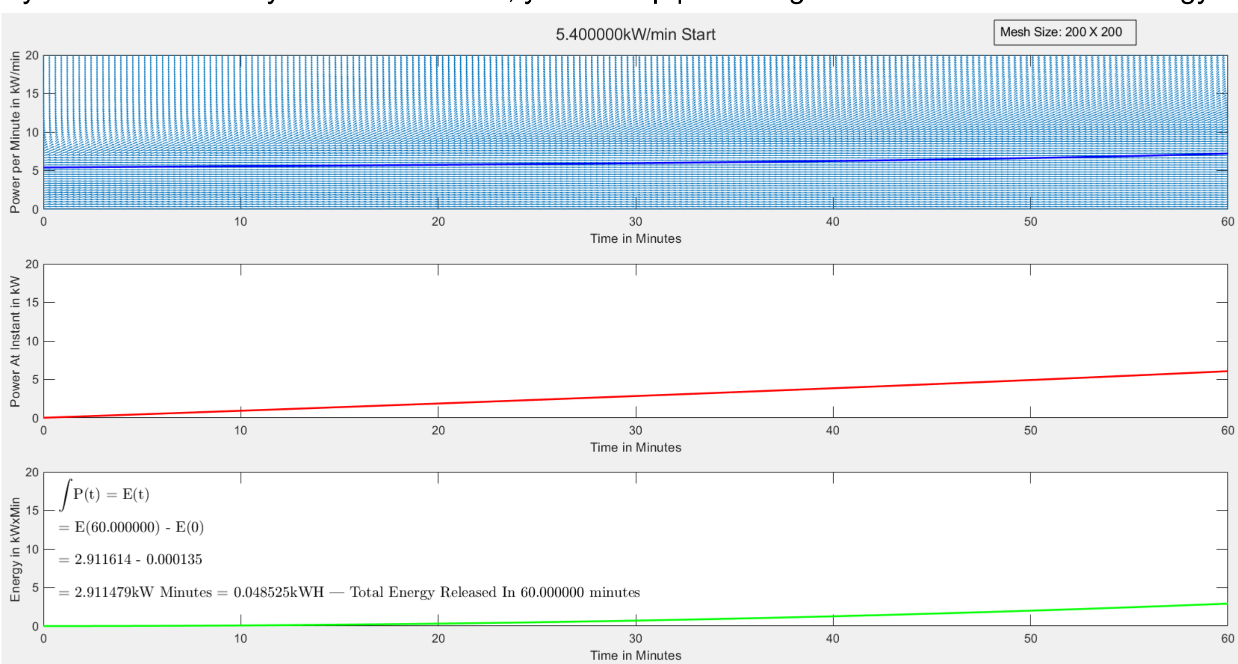
```
%%Defining Mesh
    increments = 200;
    minutes = 60;
    start = 1;
    meshy = linspace(0,minutes,increments);
    [t,P] = meshgrid(meshy,meshy);
    Graphs = tiledlayout(3,1);
    Graphs.Title.String = sprintf("%f",start)+"kW/min Start";
    annotation('textbox', [0.75, 0.875, 0.1, 0.1], 'String', "Mesh
 Size: " + increments + " X " + increments)
    maxY = 20;
    textOffSet = 3;
%%Explanation:
    %We are only given the rate of the generated power.
    %Integrating once would give you the power output at each point
    %Integrating twice would give the total power generated after an
 hour

%%Rate at which power is generated:
    %Calculating
        PowerRate = (P.*(P-5).^2)./(50+P.*t);
        L = 1./(sqrt(1+PowerRate.^2));
        M = PowerRate./(sqrt(1+PowerRate.^2));
    %Plotting
        nexttile
        quiver(t,P,L,M);
        axis on
        axis([0,minutes 0,maxY]);
        xlabel("Time in Minutes")
        ylabel("Power per Minute in kW/min")
        hold on
        SlopeRate = @(t,P)(P*(P-5)^2)/(50+P*t);
        [t,P] = ode45(SlopeRate,meshy,start);
        plot(t,P,'b','LineWidth',1.5);
        hold off

%%Power at each instance
    %Calculating
        sol = ode45(SlopeRate,meshy,start);
        PowerSlope = deval(sol,t);
        PowerInstance = [PowerSlope(1)];
        for i = 1:length(t)-1
            PowerInstance(i+1) = PowerInstance(i) + PowerSlope(i+1);
        end
        PowerInstance = PowerInstance/increments;
    %Plotting
        nexttile
        plot(t,PowerInstance,'r','LineWidth',1.5)
        axis([0,minutes 0,maxY]);
        xlabel("Time in Minutes")
        ylabel("Power At Instant in kW")
```

```
%%Total Power Up To Time
    %Calculating
        PowerSum = [PowerInstance(1)];
        for ii = 1:length(t)-1
            PowerSum(ii+1) = PowerSum(ii) + PowerInstance(ii+1);
        end
        PowerSum = PowerSum/increments;
    %Plotting
        nexttile
        plot(t,PowerSum,'g','LineWidth',1.5)
        axis([0,minutes 0,maxY]);
        xlabel("Time in Minutes")
        ylabel("Energy in kWxMin")

        text(0.75,maxY-textOffSet,"$$ \int$$P(t) =
 E(t)",'Interpreter','latex','FontSize',12);
        text(0.75,(maxY-textOffSet)*0.75,sprintf("= E(%f) -
 E(0)",minutes),'Interpreter','latex','FontSize',12);
        text(0.75,(maxY-textOffSet)*0.5,sprintf("= %f -
 %f",PowerSum(end),PowerSum(1)),'Interpreter','latex','FontSize',12);
        text(0.75,(maxY-textOffSet)*0.25,sprintf("=
 %fkW Minutes = %fkWH | Total Energy Released In %f
 minutes",(PowerSum(end) - PowerSum(1)),(PowerSum(end) -
 PowerSum(1))/60,minutes),'Interpreter','latex','FontSize',12);
```

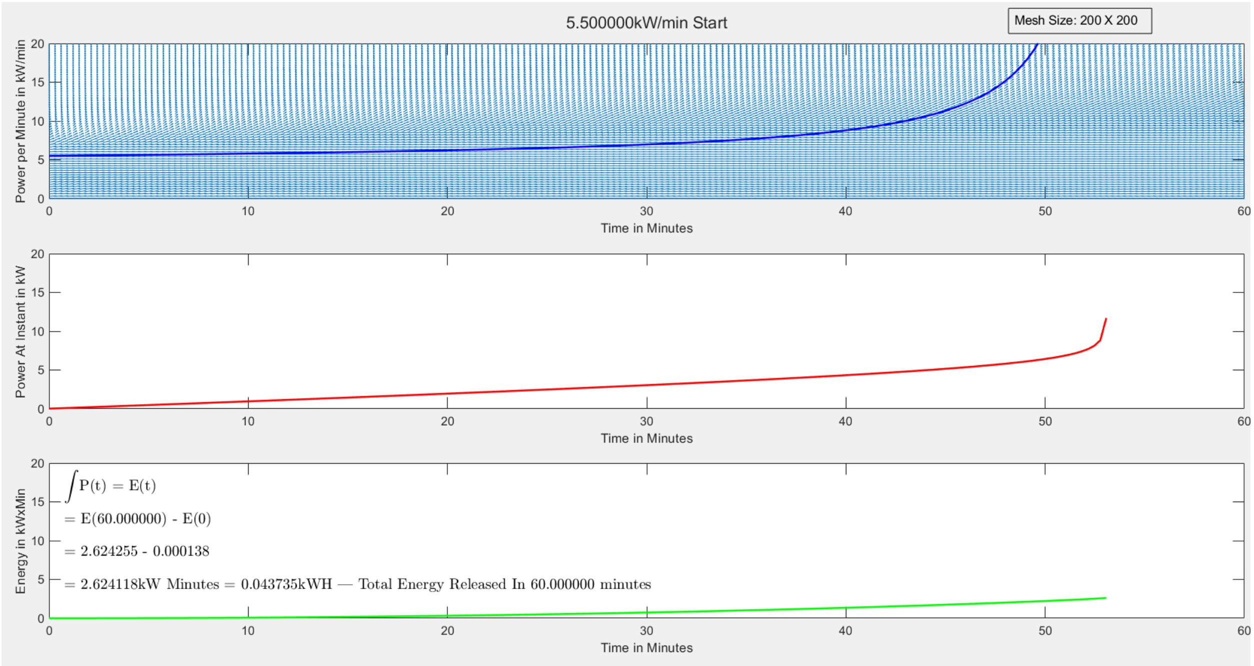
RESULTS AND EXPLANATION



If you start the battery at 1kW/min at 0s, you end up producing about 0.029563kWH of energy



The Differential Equation seems to asymptote somewhere between P = 5kW/min, and P = 5.5kW/min. This means that it produces a near constant flow of power. I believe that 5.4kW/min is the best power to calibrate the battery at, and after an hour, the battery will have outputted about 0.048525kWH. At, and past 5.5kW, the graph seems to spike.




```
>> CalcConsultationSLOPEFIELDS
Warning: Failure at t=5.315817e+01.  Unable to meet integration tolerances without reducing the step
size below the smallest value allowed (1.136868e-13) at time t.
> In ode45 (line 360)
In CalcConsultationSLOPEFIELDS (line 31)
Warning: Failure at t=5.315817e+01.  Unable to meet integration tolerances without reducing the step
size below the smallest value allowed (1.136868e-13) at time t.
> In ode45 (line 360)
In CalcConsultationSLOPEFIELDS (line 37)
```

Accepted Answer

Star Strider MVP on 15 May 2015 Vote 0 Link

I would *increase* the tolerance (to perhaps **1E-8** or more) rather than decrease it, but that may not solve your problem.

The warning is that at `t=30`, your ODE is most likely encountering a singularity. First, plot it to see what it is doing, then start it again at some value greater than `t=30` (perhaps `t=35`) and see what it does, then experiment further to get the result you want. You may have to do a piecewise integration. That may not work in the end (you may not be able to integrate it beyond `t=30` regardless), but unless you experiment with it, you will never know.

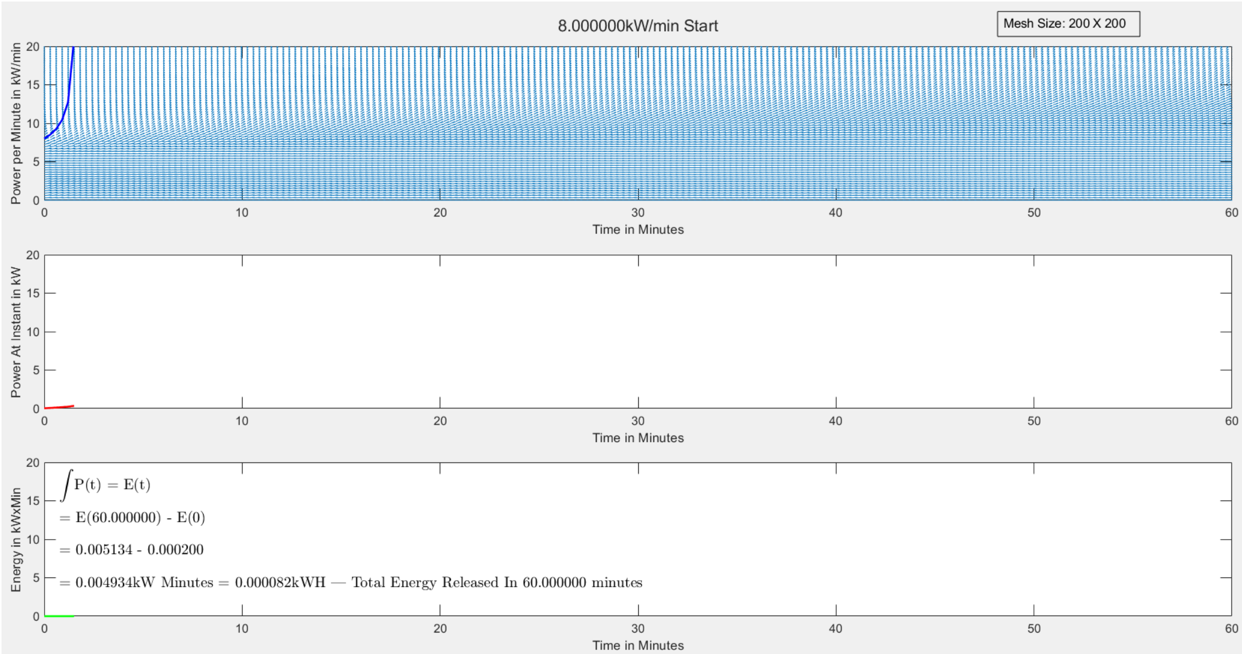
6 Comments Show 5 older comments

Star Strider MVP on 15 May 2015

Neither did Fred.

Sign in to comment.

At 5.5kW/min, the Differential Equation seems to shoot off towards positive infinity. This is why all of the graphs seem to abruptly cut off around 50 minutes. That is when the rate hits some sort of “singularity”, as Star Strider from the Matlab Forums described. This means that the rate hits a vertical asymptote at around t = 50 minutes. The error Matlab returns says t=5.315817E+01, which is about 53.15817 minutes.



Past 5.5kW/min, the energy produced means nothing. The power flow shoots off to infinity. This surge of infinite power will seriously damage the battery, and I would assume that this would be one of the “catastrophic effects” mentioned in the paper.

Unlike what your engineers have suggested, you will not be able to calibrate the battery at 8kW/min, because this would result in the aforementioned spike in power.

You can interpret the graphs cutting off as the batteries exploding, or dying, and therefore, not being able to supply any more power.

Sources:

- Matlab Forum Post: https://www.mathworks.com/matlabcentral/answers/216913-ode-problem-unable-to-meet-integration-tolerances