```
function [Thrust, M, T] = getBurnerThrust(angles)

% Designated fuel flow rate
dmdot_dt = @(x) 1 * ((x <= 0.5) * sin(pi * x) + ...
    (x > 0.5) * (x <= 2.5) * 1 + ...
    (x > 2.5) * (x <= 3) * sin(pi * (x - 2))) + ...
    (x > 3) * 0;

% Operating at 20 km
Pa = 5474.89; % [Pa] Ambient Pressure
Ta = 216.65; % [K] Ambient Temperature
burner = aae550.final.Burner();
burner.setMaxStep(1e-2);

% Set up the geometry
w = 1.067724; % need to calculate this
h = w / 5;

numSegments = numel(angles);
totalLength = 3;
width = w;
height = h;
lengths = ones(1, numSegments) * totalLength / numSegments;


% Setup the initial flow
M0 = 5; % Freestream mach
M3 = M0 / 3; % Mach at isolator exit
pr = 0.3; % Inlet/compression system total pressure recovery factor
mdot = 100; % [kg/s] Mass flow of air at isolator exit
h = 120908000;  % J/kg
startFlow = aeroBox.flowFields.FlowElement();
startFlow.setCp(1216); % J/kg*K
startFlow.setR(287.058); % J/kg*K

startFlow.setGamma(1.4);
startFlow.setMach(M3);
startFlow.setStagnationTemperature(aeroBox.isoBox.calcStagTemp('mach',
 M0, 'gamma', 1.4, 'Ts', Ta));
startFlow.setStagnationPressure(aeroBox.isoBox.calcStagPressure('mach',
 M0, 'gamma', 1.4, 'Ps', Pa) * pr);
startFlow.setMassFlow(mdot);

cea = nasa.CEARunner();
params = cea.run('prob', 'tp', 'p(bar)', startFlow.P()/1e5, 't,k',
 startFlow.T(), 'reac', 'name', 'Air', 'wt%', 100, 'end');

startFlow.setGamma(params.output.gamma);
startFlow.setCp(params.output.cp * 1e3);

burner.setGeometry(width, height, lengths, angles);
burner.setHeatingValue(h);
```

```
burner.setInjectionFunc(dmdot_dt);
burner.setStartFlow(startFlow);


% Setup solver


burner.solve();

states = burner.states;
M = zeros(1, numel(states));
for l = 1:numel(states)
     x(l) = states{l}.x;
    flow = states{l}.flow;
    M(l) = flow.M();
     mdot(l) = flow.mdot();
%    u(l) = flow.u();
%    Pt(l) = flow.Pt();
%    Tt(l) = flow.Tt();
%    R(l) =flow.R();
%    cp(l) = flow.cp();
    T(l) = flow.T();
%   P(l) = flow.P();
%    gamma(l) = flow.gamma();
end
if any(M < 1)
    Thrust = 1;
else
    endFlow = burner.states{end}.flow;
    Me = aeroBox.isoBox.machFromPressureRatio('Prat', Pa /
 endFlow.Pt, 'gamma', endFlow.gamma);
    ue = Me * endFlow.getSonicVelocity();
    Thrust = ue * endFlow.mdot();
end

shouldPlot = 1;
if shouldPlot
    figure;
    subplot(1, 2, 1);
    plot(x, M);
    xlabel('Distance (m)');
    ylabel('Mach Number');
    title('Mach Number vs. Distance');

    subplot(1, 2, 2);
    plot(x, T);
    xlabel('Distance (m)');
    ylabel('Static Temperature (K)');
    title('Static Tempterature vs. Distance');

    burner.plotGeometry();
    axis equal
end
end
```

Published with MATLAB® R2015b