```
classdef FlowElement < handle
    %FLOW Basic flow element
    % Created by Thomas Satterly
    properties (SetAccess = private)
        gamma; % Ratio of specific heats
        cp; % Specific heat at constant pressure
        R; % Gas constant
        Tt; % Stagnation temperature
        Pt; % Stagnation pressure
        rho_t; % Stagnation density
        M; % Mach number
        mdot; % Mass flow of stream
    end

    methods

        function fe = getCopy(obj)
            % Returns a deep copy of the flow element
            feh = getByteStreamFromArray(obj);
            fe = getArrayFromByteStream(feh);
        end

        function t = T(obj)
            % Returns the static temperature of the flow
            t = aeroBox.isoBox.calcStaticTemp('mach', obj.M, 'Tt',
 obj.Tt, 'gamma', obj.gamma);
        end

        function setCp(obj, cp)
            obj.cp = cp;
        end

        function setGamma(obj, gamma)
            obj.gamma = gamma;
        end

        function setR(obj, R)
            obj.R = R;
        end

        function setStagnationTemperature(obj, t)
            % Sets the stagnation temperature
            obj.Tt = t;
        end

        function setStaticTemperature(obj, t)
            % Sets the flow properties to match the desired static
 temperature
            obj.Tt = aeroBox.isoBox.calcStagTemp('mach', obj.M, 'Ts',
t, 'gamma', obj.gamma);
        end
```

```
        function p = P(obj)
            % Returns the static pressure of the flow
            p = aeroBox.isoBox.calcStaticPressure('mach', obj.M, 'Pt',
obj.Pt, 'gamma', obj.gamma);
        end

        function setStagnationPressure(obj, p)
            % Sets the stagnation pressure
            obj.Pt = p;
        end

        function setStaticPressure(obj, p)
            % Sets the flow properties to match the desired static
pressure
            obj.Tt = aeroBox.isoBox.calcStagPressure('mach',
obj.M, 'Ps', p, 'gamma', obj.gamma);
        end

        function r = rho(obj)
            % Returns the static density
            r = aeroBox.isoBox.calcStaticDensity('mach',
obj.M, 'rho_t', obj.rho_t, 'gamma', obj.gamma);
        end

        function setStagnationDensity(obj, r)
            obj.rho_t = r;
        end

        function setStaticDensity(obj, r)
            obj.rho_t = aeroBox.isoBox.calcStagDensity('mach',
obj.M, 'rho', r, 'gamma', obj.gamma);
        end

        function m = u(obj)
            % Returns the mach number of the flow
            m = obj.M * obj.getSonicVelocity();
        end

        function setMach(obj, m)
            % Sets flow properties to match the desired mach number
            obj.M = m;
        end

        function a = getSonicVelocity(obj)
            % Returns the sonic velocity of the flow
            a = sqrt(obj.gamma * obj.R * obj.T());
        end

        function a = getArea(obj)
            % Returns the area of the flow
            a = obj.A;
        end

        function setMassFlow(obj, mdot)
```

```
            obj.mdot = mdot;
        end

%         function setMassFlow(obj, mdot, variable)
%             switch variable
%                 case 'density'
%                     rho = mdot / (obj.u * obj.A);
%                     obj.rho_t =
 aeroBox.isoBox.calcStagDensity('mach', obj.M, 'rho', rho, 'gamma',
 obj.gamma);
%                 case 'velocity'
%                     obj.u = mdot / (obj.rho() * obj.A);
%                 case 'area'
%                     obj.A = mdot / (obj.rho() * obj.u);
%                 otherwise
%                     error('Invalid input variable ''%s''',
 variable);
%             end
%         end
    end

end
```

Published with MATLAB® R2015b