```
function [r,v] = oe2states(h,e,RA,incl,w,TA,mu)
%OE2STATES Converts orbital elements to state vector
%   Inputs are:
%   h       :a scalar specific angular momentum in m^2/s
%   e       :a scalar orbital eccentricy
%   RA      :a scalar right ascension of the ascending node in rad
%   incl    :a scalar orbital inclination
%   W       :a scalar argument of perigee in rad
%   TA      :a scalar true anomaly in rad
%   mu      :an optional scalar gravitational parameter in m^3/s^2 (default
%            earth)
%
%   Outputs are:
%   r       :a numeric array of 3x1 current position vector in m
%   v       :a numeric array of 3x1 current velocity vector in m/s

    arguments
        h {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        e {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        RA {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        incl {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        w {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        TA {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        mu {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal} = 3.98600442e14
    end

%   r and v in perifocal frame
    rp = (h^2/mu)*(1/(1 + e*cos(TA)))*[cos(TA);sin(TA);0];
    vp = (mu/h)*[-sin(TA);e+cos(TA);0];

%   Rotate perifocal coordinates to geocentric equatorial frame
    Q = euler_rot(RA,incl,w,'313')';
    r = Q*rp;
    v = Q*vp;
end
```