```matlab
function [a, e, i, RAAN, omega, M] = cart2kep(r,v,mu)
%-------------------------------------------------------------------------
% Converts ECI state orbit to Orbital elements
%-------------------------------------------------------------------------
% Inputs: r and v [m] and [m/s] of orbit
%         mu  [m^3/s^2]
%
% Outputs: a [m] - semi major axis
%          e  - eccentricity
%          i [deg] inclination
%          RAAAN [deg] Right angle of ascending node
%          omega [deg]  argument of perigee
%          M [deg]  mean anomaly
%
%
    h = cross(r,v);
    h_mag = norm(h);
    n = cross([0;0;1],h);
    n_mag = norm(n);
    e_vec = cross(v,h)/mu - r/norm(r);
    e = norm(e_vec);
    a = 1/(2/norm(r) - norm(v)^2/mu);
    i = acos(h(3)/h_mag);
    RAAN = atan2(h(1), -h(2));
    omega = atan2(dot(cross(n,e_vec),h)/h_mag, dot(n,e_vec));
    nu = atan2(dot(cross(e_vec,r),h)/h_mag, dot(e_vec,r));

    if e < 1 - 1e-12         % elliptic
        % Compute eccentric anomaly E robustly from nu
        denom = 1 + e * cos(nu);
        cosE = (e + cos(nu)) ./ denom;
        sinE = sqrt(max(0, 1 - e^2)) .* sin(nu) ./ denom;
        E = atan2(sinE, cosE);
        E = mod(E, 2*pi);
        M = E - e .* sin(E);
        M = mod(M, 2*pi);    % normalized
    elseif e > 1 + 1e-12    % hyperbolic
        % hyperbolic anomaly F (sometimes H)
        denom = 1 + e * cos(nu);
        if abs(denom) < eps
            warning('Denominator near zero when converting nu to hyperbolic anomaly;↙
result may be unstable.');
        end
        sinhF = sqrt(e^2 - 1) .* sin(nu) ./ denom;
        F = asinh(sinhF);
        M = e .* sinh(F) - F;   % hyperbolic "mean anomaly" (signed)
        % do not normalize M to [0,2pi) for hyperbola
    else
        error('Parabolic orbit (e ~ 1): mean anomaly is not defined; use Barker''s↙
equation for time-of-flight.');
    end

    kep = [a; e; i; RAAN; omega; M];
```

```
end
```

```
end
```