```
function [r,v] = RVFromCOE( a,i,W,w,e,th, mu )

% Compute position and velocity vectors in the geocentric equatorial
 frame
% from the common orbital elements
%
% All anges are in radians, all distances in km.
%
%   Inputs:
%     a       Semi major axis
%     i       Inclination
%     W       Right ascension of ascending node
%     w       Argument of perigee
%     e       Eccentricity
%     th      True anomaly
%     mu      Graviational constant
%
%   Outputs:
%     r       (3,1)   Position vector in inertial frame
%     v       (3,1)   Velocity vector in inertial frame

if( nargin==2 )
  el = a;
  mu = i;
  [a,i,W,w,e,th] = OrbitalElements(el);
end

% specific angular momentum
h = sqrt(mu*abs(a*(1-e^2)));

% we may have more than one true anomaly "th"
n = length(th);

c = cos(th);
s = sin(th);

% first compute the position and velocity in perifocal frame
rp = h^2/mu*[ c./(1+e*c); s./(1+e*c); zeros(1,n) ];
vp = mu/h*[ -s; e+c; zeros(1,n) ];

% Compute the rotation matrix that rotates vectors:
%   FROM Geocentric/Equatorial
%     TO Perifocal
Q = GeoEqToPerifocal(i,W,w);

% transform from the perifocal to Geocentric Equatorial frame
% going REVERSE direction, so transpose Q
r = Q'*rp;
v = Q'*vp;
```

Published with MATLAB® R2019b