AE4878 - Mission Geometry and Orbit Design
Part 1 - Coordinate Transformations

Ali Nawaz
3ME, LR, Delft University of Technology.

September 2, 2018

1 Introduction

The aim of this assignment is to write a software routine capable of converting Cartesian component to Kepler Elements and vice versa for elliptical/circular orbit. Section 1.1 outlines the coordinate systems used to facilitate the conversions, along with a list of symbols with their representative values where necessary. Chapter 2 outlines the transformation of Cartesian Component to Kepler Elements, while chapter 3 outlines the transformation of Kepler Elements to Cartesian Components.

1.1 Co-ordinate System and Parameters Values

Figure 1 outlines the three dimensional Kepler co-ordinate frame used throughout this assignment for transformation[1]. Figure 2 [2] outlines the intermediate reference frame used to convert Kepler elements to Cartesian component.Elements in Figure 1 are outlined in Table 1

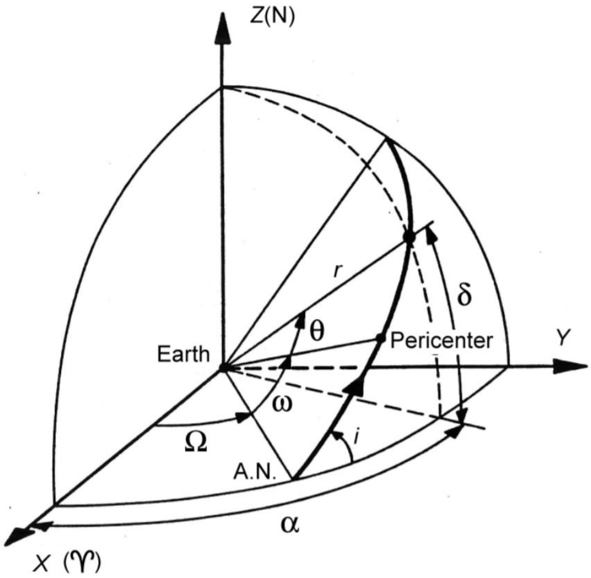

Figure 1: Three dimensional Kepler coordinate frame.

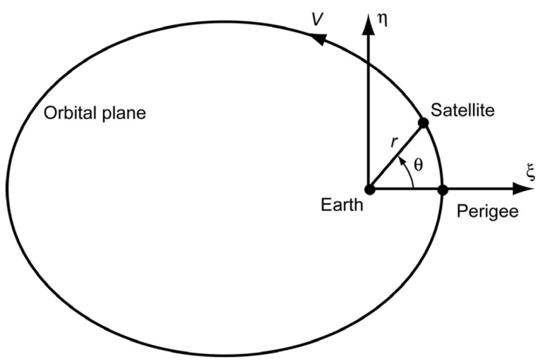

Figure 2: Coordinate for positional transformation from Kepler to Cartesian

Figure 3 [3] illustrates geocentric and geodetic coordinate frame for Kepler two body problem. Throughout this assignment geocentric coordinate frame is used.

[1] Fundamentals of Astrodynamics - K. Wakker
pg.261 https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics
[2] Fundamentals of Astrodynamics - K. Wakker
pg.269 https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics
[3] Fundamentals of Astrodynamics - K. Wakker
pg.263 https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics

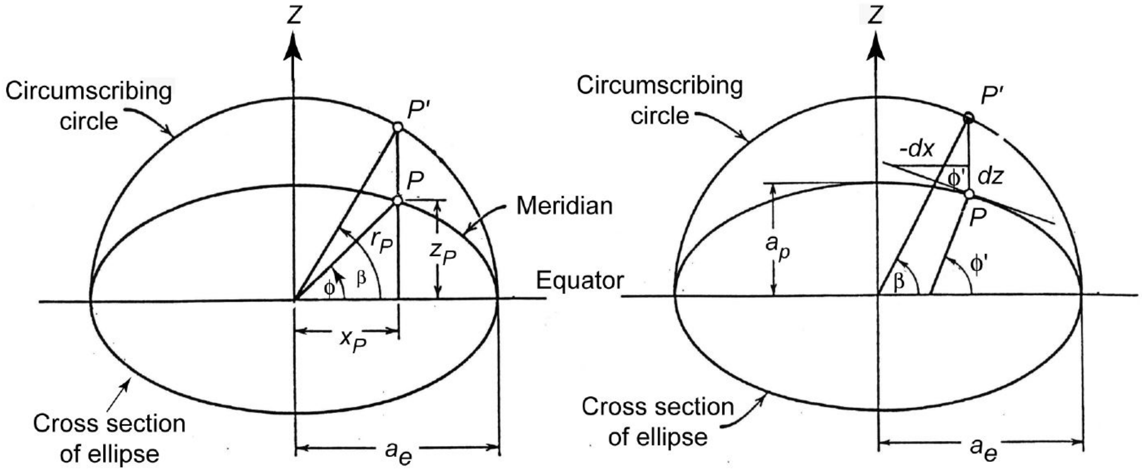

Figure 3: Ellipsoid cross section for geocentric and geodetic reference frames. Throughout this assignment geocentric frame is used.

Table 1[4] outlines the elements from Figures 1-3 used for this assignment. For detailed description please refer to the cited sources.

Table 1: Information for parameters used in the assignment.

Parameter	Nomenclature	Unit	Value
n	Mean motion	rad/s	-
θ	True anomaly	rad	-
E	Eccentric anomaly	rad	-
M	Mean anomaly	rad	-
e	eccentricity	-	-
a	Semi major axis	m	-
t	time passage of perigee	s	-
r	orbital position vector	m	-
i	inclination angle	rad	-
Ω	Right ascension of the textbackslash ascending node	rad	-
ω	Argument of pericenter	rad	-
H	Angular momentum	$\frac{m^2}{s}$	-
V	Cartesian velocity vector	m/s	-
N	Vector to ascending node	m	-
μ	Standard gravitational parameter, Earth	$\frac{m^3}{s^2}$	3.986004419$\cdot 10^{14}$

2 Cartesian Component to Kepler Elements

The aim of this section is to go from Cartesian [x, y, z $\dot{x}, \dot{y}$, $\dot{z}$]' to Kepler Elements [a, e, i, Ω, ω, θ, E, M]'. Nomenclature for these parameters are outlined in Section 1. In the following subsections, first the problem is formulated in Matlab. Data presented in the lectures is used to verify the results produced by the Matlab script. To conclude, final values obtained for the assignment is outlined.

[4] Earth's standard gravitational parameter, μ `https://en.wikipedia.org/wiki/Standard_gravitational_parameter`

2.1 Problem Formulation

Position and velocity vectors in the Cartesian co-ordinate frame are defined as outlined in Equation 1.

$$\begin{aligned}&\text{Position vector } \bar{r}\text{:}\\ &\bar{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}\\ &\text{Velocity vector } \bar{V}\text{:}\\ &\bar{V} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}\end{aligned} \tag{1}$$

This leads to the formulation of angular momentum vector $\bar{h}$ and vector to the ascending node $\bar{N}$. This is outlined in Equation 2.

$$\begin{aligned}&\text{The angular momentum vector } \bar{h}\text{:}\\ &\bar{h} = \bar{r} \times \bar{V}\\ &\text{Vector to the ascending node } \bar{N}\text{:}\\ &\bar{N} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \bar{h}\end{aligned} \tag{2}$$

With the above information, semi major axis (a), inclination angle (i) and eccentricity vector ($\bar{e}$) can be found using Equation 3.

$$\begin{aligned}&\text{Semi major axis a:}\\ &a = \frac{1}{\frac{2}{|\bar{r}|} - \frac{|\bar{V}|^2}{\mu}}\\ &\text{The eccentricity vector, } \bar{e}\text{:}\\ &\bar{e} = \frac{\bar{V} \times \bar{h}}{\mu} - \frac{\bar{r}}{|\bar{r}|}\\ &\text{ The inclination angle can be found using:}\\ &i = acos(\frac{h_z}{|\bar{h}|}\end{aligned} \tag{3}$$

The vector to the ascending node $\bar{N}$ can be simplified to obtain the right ascension of ascending node Ω, argument of pericenter ω and true anomaly θ. This is shown with the aid of Equation 4.

$$\begin{aligned}&\text{From the vector of ascending following terms can be deduced:}\\ &N_{xy} = \sqrt{N_x^2 + N_y^2}\\ &\text{This results in the following relation for } \Omega\text{:}\\ &\Omega = atan2(\frac{N_y}{N_{xy}}, \frac{N_x}{N_{xy}})\\ &\text{Formulating relations for } \omega \text{ and } \theta \text{ includes a logical statement for choice of value:}\\ &\omega = sign \cdot acos(\frac{\bar{e} \cdot \bar{N}}{|\bar{e}| \cdot |\bar{N}|}\\ &\text{Where, } sign = +1 \text{ if } (\bar{N} \times \bar{e}) \cdot \bar{h} > 0\text{; -1 otherwise}\\ &\theta = sign \cdot acos(\frac{\bar{r} \cdot \bar{e}}{|\bar{r}| \cdot |\bar{e}|})\\ &\text{Where, } sign = +1 \text{ if } (\bar{e} \times \bar{r}) \cdot \bar{h} > 0\text{; -1 otherwise}\end{aligned} \tag{4}$$

Once θ is obtained, the formulation outlined in Equation 5 leads to finding eccentric anomaly E and mean anomaly M. Note the relation is valid for a $0 \leq e < 1$

$$tan(\frac{\theta}{2}) = \sqrt{\frac{1+|\bar{e}|}{1-|\bar{e}|}} \cdot tan(\frac{E}{2})$$
$$M = E - e \cdot sin(E) \tag{5}$$

All the above relations are derived from the cited sources[5] [6] [7]. Following outlines the Matlab script used to conduct the transformations:

```
%% Part 1 : Convert state vector from Cartesian to Kepler elements
% Actual values
x = 8751268.4691; % [m]
y = -7041314.6869; % [m]
z = 4846546.9938; % [m]

xdot = 332.2601039; % [m/s]
ydot = -2977.0815768; % [m/s]
zdot = -4869.8462227; % [m/s]
% % Verification values
% x = -2700816.14; % [m]
% y = -3314092.80; % [m]
% z = 5266346.42; % [m]

% xdot = 5168.606550; % [m/s]
% ydot = -5597.546618; % [m/s]
% zdot = -868.878445; % [m/s]

r_vec = [x;y;z]; % position vector [m/s]

V_vec = [xdot; ydot; zdot]; % velocity vector [m/s]

r = norm(r_vec); % position norm [m]
V = norm(V_vec); % velocity norm [m/s]

h = cross(r_vec,V_vec); % angular momentum [m^2/s]
N = cross([0; 0; 1],h); % vector to ascending node [m]

a = 1/ ((2/r) - ( (V^2)/(mu) )); % semi major axis [m]

e_vec = (1/mu)*cross(V_vec,h) - (1/r)*r_vec ; % Eccentricity vector
e = norm(e_vec); % norm of eccentricity

i = acos(h(end)/ norm(h)); % inclination angle [rad]
i_deg = rad2deg(i); % inclination angle [deg]

Nx = N(1); % x component of N
Ny = N(2); % y component of N

Nxy = sqrt(Nx^2 + Ny^2); % norm of x.y component of N

Omega = atan2( Ny/Nxy, Nx/Nxy); % Right ascension of ascending node [rad]
Omega_deg = rad2deg(Omega); % Right ascension of ascending node [deg]

if dot(cross(N,e_vec),h)>0
    sign_omega = 1;
else
    sign_omega = -1;
end

omega = sign_omega* (acos( dot( e_vec/e, N/norm(N)))); % argument of pericenter [rad]
omega_deg = rad2deg(omega); % argument of pericenter [deg]

if dot(cross(e_vec,r_vec),h)>0
    sign_theta = 1;
else
    sign_theta = -1;
end
```

[5]Orbit Constellation Design Management, 2nd printing- Wertz, James R.

[6]Fundamentals of Astrodynamics - K Wakker
https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics

[7]AE4878 Mission Geometry and Orbit Design Lecture Slides - R. Noomen, TU Delft

```

theta = sign_theta* acos(dot(r_vec/r,e_vec/e)); % True anomaly [rad]
theta_deg = rad2deg(theta); % True anomaly [deg]

E = 2*atan( sqrt((1-e)/(1+e))* tan(theta/2)); % Eccentric anomaly [rad]
E_deg = rad2deg(E); % Eccentric anomaly [deg]

M = E- e*sin(E); % Mean anomaly [rad]
M_deg = rad2deg(M); % Mean anomaly [deg]

% % Percentage offset for verification
% a_off = (abs(6787746.891 - a)/(6787746.891) )*100; % Offset semi major axis [m]
% e_off = (abs(0.000731104 - e)/(0.000731104) )*100; % Offset eccentricity [-]
% i_off = (abs(51.68714486 - i_deg)/(51.68714486) )*100; % Offset inclination angle [deg]
% Omega_off = (abs(127.5486706 - Omega_deg)/(127.5486706) )*100; % Offset Right Ascension of ...
    Ascending Node [deg]
% omega_off = (abs(74.21987137 - omega_deg)/(74.21987137) )*100; % Offset omega [deg]
% theta_off = (abs(24.10027677 - theta_deg)/(24.10027677 ) )*100; % Offset theta [deg]
% E_off = (abs(24.08317766 - E_deg)/(24.08317766) )*100; % Offset Eccentric anomaly [deg]
% M_off = (abs(24.06608426 - E_deg)/(24.06608426) )*100; % Offset Mean anomaly [deg]

offset = [a_off, e_off, i_off, Omega_off, omega_off, theta_off, E_off, M_off]; % Vector of ...
    offset between output and slide values for verification stage
```

2.2 Verification

The above script is first verified with known values presented in the lecture slide [8]. Table 2 outlines the input to the above script in Cartesian form, and the corresponding output in Kepler elements, compared with the results outlined in the lecture slides. The Table indicates that the output results from the the script are verified to be correct. Correct in the sense that they are very close to the values presented in the lecture slides, however it is important to note that for increasing decimal places the value are not exactly same for some parameters.

Table 2: Cartesian input elements and their corresponding Kepler output elements. For verification purposes values from the lecture slides is presented. The output values are presented upto 10 significant figures.

Input Element	Input Value	Unit	Output Element	Output Value	Verification Value	% Offset	Unit
x	-2700816.14	[m]	a	6.787746876e+06	6787746.891	2.261024896e-07	m
y	-3314092.80	[m]	e	7.311020662e-04	0.000731104	2.645074976e-04	-
z	5266346.42	[m]	i	51.68714486	51.68714486	8.318441965e-10	deg
V_x	5168.606550	[m/s]	Ω	127.5486706	127.5486706	1.954917023e-08	deg
V_y	-5597.546618	[m/s]	ω	74.21979912	74.21987137	9.734967710e-05	deg
V_z	-868.878445	[m/s]	θ	24.10034902	24.10027677	2.998057252e-04	deg
			E	24.08324991	24.08317766	3.000200981e-04	deg
			M	24.06615651	24.06608426	0.07132715978	deg

2.3 Final results

To conclude this Section, Table 3 outlines transformation from Cartesian to Kepler Elements for the Cartesian values presented in Assignment Basics 1.

Table 3: Final results for Basic 1, question 1, upto 10 significant figures.

Input Element	Input Value	Unit	Output Element	Output Value	Unit
x	8751268.4691	[m]	a	1.227308615e+07	m
y	-7041314.6869	[m]	e	0.005022165232	-
z	486546.9938	[m]	i	1.098187738e+02	deg
V_x	332.2601039	[m/s]	Ω	1.322336978e+02	deg
V_y	-2977.0815768	[m/s]	ω	1.050667132e+02	deg
V_z	-4869.8462227	[m/s]	θ	50.02801109	deg
				E	49.80784631	deg
			M	49.58803943	deg

[8] AE4878 Mission Geometry and Orbit Design Lecture- Basics Slide - 20, - R. Noomen, TU Delft

3 Kepler Elements to Cartesian Components

The aim of this section is to go from Kepler Elements [a, e, i, Ω, ω, θ, E, M]' to Cartesian components [x, y, z $\dot{x}, \dot{y}$, $\dot{z}$]'. Nomenclature for these parameters are outlined in Section 1. In the following subsections, first the problem is formulated in Matlab. Data presented in the lectures is used to verify the results produced by the Matlab script. To conclude, final values obtained for the assignment is outlined.

3.1 Problem Formulation

It can be observed from values in Table 4 that $0 \leq |\bar{e}| < 1$. Eccentric anomaly for such an orbit can be found iteratively using the value of mean anomaly and eccentric anomaly. E is iterated to a final value as outlined in Equation 6. Initial value for E is chosen as M and a tolerance is set to a desired level to obtain a desired degree of convergence to the correct value of E. This is treated in the script presented.

$$E_{i+1} = E_i + \frac{M - E_i + e \cdot sin(E_i)}{1 - e \cdot cos(E_i)} \tag{6}$$

Once the Eccentric anomaly is obtained, the relations outlined in Equation 7 is used to obtain true anomaly θ and r, the norm of position vector $\bar{r}$.

$$\begin{aligned}
&\text{True anomaly } \theta \text{ is obtained using:}\\
&\tan(\frac{\theta}{2}) \quad = \quad \sqrt{\frac{1+e}{1-e}} \quad \cdot \quad \tan(\frac{E}{2})\\
&\text{Note e} = |\bar{e}|\\
&r = a(1 - e \cdot \cos(E))
\end{aligned} \tag{7}$$

θ and r, obtained above can be used to express the coordinate frame $(\mathcal{E}, \eta)$ in Figure 2. Where, $\mathcal{E} = r \cdot cos(\theta)$ and $\eta = r \cdot sin(\theta)$. This can be used to facilitate a transformation matrix, which produces [x, y, z] components in the Cartesian frame. This is outlined in Equation 8 [9].

$$\begin{aligned}
&\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1 & l_2 \\ m_1 & m_2 \\ n_1 & n_2 \end{bmatrix} \begin{bmatrix} \mathcal{E} \\ \eta \end{bmatrix}\\
&\text{where,}\\
&l_1 = \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i\\
&l_2 = -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i\\
&m_1 = \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i\\
&m_2 = -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i\\
&n_1 = \sin\omega\sin i\\
&n_2 = \cos\omega\sin i
\end{aligned} \tag{8}$$

The magnitude of angular moment, H of the system, is found using the relation outlined in Equation 9. Finally this magnitude of angular moment is used to find the velocity components in the Cartesian co-ordinate frame. This is demonstrated with the aid of Equation 9.

$$\begin{aligned}
&H = \sqrt{\mu \cdot a \cdot (1 - e^2)}\\
&\dot{x} = \frac{\mu}{H}[-l_1 \sin\theta + l_2(e + \cos\theta)]\\
&\dot{y} = \frac{\mu}{H}[-m_1 \sin\theta + m_2(e + \cos\theta)]\\
&\dot{z} = \frac{\mu}{H}[-n_1 \sin\theta + n_2(e + \cos\theta)]
\end{aligned} \tag{9}$$

[9] AE4878 Mission Geometry and Orbit Design Lecture- Basics, Slides 17 - R. Noomen, TU Delft

All the above relations are derived from the cited sources[10] [11] [12]. Following outlines the Matlab script used to conduct the transformations:

```matlab
% Part 2 : Convert state vector from Kepler elements to Cartesian components
% Actual values for the assignment
a = 12158817.9615; % Semi major axis [m]
e = 0.014074320051; % [eccentricity]
i =  deg2rad(52.666016957); % inclination angle in rad
Omega = deg2rad(323.089150643); % Right Ascension of Ascending Node [rad]
omega = deg2rad(148.382589129); % Argument of pericenter [rad]
M = deg2rad(112.192638384); % Mean anomaly [rad

% % Verification values
% a = 6787746.891;
% e = 0.000731104;
% i = deg2rad(51.68714486);
% Omega = deg2rad(127.5486706);
% omega = deg2rad(74.21987137);
% % theta = deg2rad(24.10027677);
% % E = deg2rad(24.08317766);
% M = deg2rad(24.06608426);

% Eccentric Anomaly E for elliptical orbit, since e <1
err = 1; % initializing difference between previous and current E
tol = 1*10^-20; % Tolerance for converge of E
E_i0 = M; % initialise E
step = 0; % Step number
while err > tol
       step = step +1;
       E_i1 = E_i0 + ( M - E_i0 + e*sin(E_i0))/( 1 - e*cos(E_i0));
       err = norm(E_i1 - E_i0);
       E_i0 = E_i1;
       E = E_i0; % Update the final value of E
end
E_deg = rad2deg(E); % Eccentric anomaly in [deg]
% Since e<1 theta follows from the kepler relation of an elliptical orbit

n = sqrt(mu/a^3); % Mean motion [rad/s]
theta = 2*atan( sqrt( (1+e)/(1-e) )*tan(E/2)); % True anomaly [rad]
theta_deg = rad2deg(theta); % True anomaly [deg]
r = a*(1-e*cos(E)); % norm of position vector [m]

% Transformation variables :
l1 = cos(Omega)*cos(omega) - sin(Omega)*sin(omega)*cos(i);
l2 = -cos(Omega)*sin(omega) -sin(Omega)*cos(omega)*cos(i);

m1 = sin(Omega)*cos(omega) + cos(Omega)*sin(omega)*cos(i);
m2 = -sin(Omega)*sin(omega) + cos(Omega)*cos(omega)*cos(i);

n1 = sin(omega)*sin(i);
n2 = cos(omega)*sin(i);

% Transformation matrix to convert Kepler position to cartesian position

% [eps,eta] is fixed on Earth, eps is the x axis and pointing to the right
% and eta the y axis point up on the orbital plane(as viewed from the top)
% Similar to Lecture slide 17 of Basic lecture.

eps = r*cos(theta);
eta = r*sin(theta);

T = [ l1 l2; m1 m2; n1 n2];
pos_cartesian = T* [eps; eta];

x = pos_cartesian(1); % x position [m]
y = pos_cartesian(2); % y position [m]
z = pos_cartesian(3); % z position [m]

% Angular momentum H
H = sqrt( mu * a*( 1 - e^2));
```

[10] Orbit Constellation Design Management, 2nd printing- Wertz, James R.

[11] Fundamentals of Astrodynamics - K Wakker https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics

[12] AE4878 Mission Geometry and Orbit Design Lecture Slides - R. Noomen, TU Delft

```
% Cartesian velocity parameters in [m/s]
x_dot = (mu/H)*( -l1*sin(theta) + l2*(e + cos(theta))); % x velocity component [m/s]
y_dot = (mu/H)*( -m1*sin(theta) + m2*(e + cos(theta))); % y velocity component [m/s]
z_dot = (mu/H)*( -n1*sin(theta) + n2*(e + cos(theta))); % z velocity component [m/s]


% % Percentage offset for verification
% x_off = (abs(-2700816.14 - x)/abs(-2700816.14) )*100; % Offset position x [m]
% y_off = (abs(-3314092.80 - y)/abs(-3314092.80) )*100; % Offset position y [m]
% z_off = (abs(5266346.42 - z)/abs(5266346.42) )*100; % Offset position z [m]
% x_dot_off = (abs(5168.606550 - x_dot)/(127.5486706) )*100; % Offset position x_dot [m/s]
% y_dot_off = (abs(-5597.546618 - y_dot)/(74.21987137) )*100;% Offset position y_dot [m/s]
% z_dot_off = (abs(-868.878445- z_dot)/(24.10027677 ) )*100; % Offset position z_dot [m/s]
% E_off = (abs(24.08317766 - E_deg)/(24.08317766) )*100; % Offset Eccentric anomaly [deg]
% theta_off = (abs(24.10027677 - theta_deg)/(24.10027677) )*100; % Offset true anomaly [deg]
% offset = [x_off, y_off, z_off, x_dot_off, y_dot_off, z_dot_off, E_off]; % Vector of offset ...
    between output and slide values for verification stage
```

3.2 Verification

The above script is first verified with known values presented in the lecture slide [13]. Table 4 outlines the input to the above script in Kepler elements, and the corresponding output in Cartesian form, compared with the results outlined in the lecture slides. The Table indicates that the output results from the script are verified to be correct. Correct in the sense that they are very close to the values presented in the lecture slides, however it is important to note that for increasing decimal places the values are not exactly the same for some parameters.

Table 4: Kepler input elements and their corresponding Cartesian output elements. For verification purposes values from the lecture slides is presented. The output values are presented upto 10 significant figures.

Input Element	Input Value	Unit	Output Element	Output Value	Verification Value	% Offset	Unit
a	6787746.891	m	x	-2.700816139e+06	-2700816.14	2.092896011e-08	[m]
e	0.000731104	-	y	-3.314092801e+06	-3314092.80	3.075290532e-08	[m]
i	51.68714486	deg	z	5.266346421e+06	5266346.42	1.288021402e-08	[m]
Ω	127.5486706	deg	V_x	5.168606557e+03	5168.606550	5.861478138e-06	[m/s]
ω	74.21987137	deg	V_y	-5.597546622e+03	-5597.546618	4.906786338e-06	[m/s]
M	24.06608426	deg	V_z	-8.688784455e+02	-868.878445	2.075441659e-06	[m/s]
			E	24.08317766	24.08317766	8.476854218e-09	deg
			θ	24.10027676	24.10027677	2.241870733e-08	deg

3.3 Final Results

To conclude this Section, Table 5 outlines transformation from Kepler Elements to Cartesian Elements for the Kepler element values presented in Assignment Basics 1.

Table 5: Final results for Basic 1, question 2, upto 10 significant figures.

Input Element	Input Value	Unit	Output Element	Output Value	Unit
a	12158817.9615	m	x	-5.760654230e+06	[m]
e	0.014074320051	-	y	-4.856967488e+06	[m]
i	52.666016957	deg	z	-9.627444862e+06	[m]
Ω	323.089150643	deg	V_x	4.187661256e+03	[m/s]
ω	148.382589129	deg	V_y	-3.797545190e+03	[m/s]
M	112.192638384	deg	V_z	-6.836151268e+02	[m/s]
			E	1.129352881e+02	deg
			θ	1.136759331e+02	deg

[13] AE4878 Mission Geometry and Orbit Design Lecture- Basics, Slide - 20, - R. Noomen, TU Delft