ESSE 3610 – Geodetic Concepts

PROJECT 1 – MILESTONE 4

October 15, 2018

Group 16

Victoria Donko 214753461
Talon Denton 214342380
Yaseen Al-Taie 213996921
Dion Farquhar 214325021

Executive Summary

This report provides an analysis and evaluation of the knowledge of theoretical and computational aspects of coordinate system transformations. It also provides insight to the computational aspects of key coordinate systems (local astronomic, apparent, celestial, orbital, and conventional terrestrial) and their transformations.

Methods of analysis include using mathematical proof of matrix transformations, as well as using an appropriate software to design, develop and apply the knowledge of these coordinate system transformations. All relevant figures and computer code can be found in their corresponding appendix. The results of Part A of this project, show that the rotation matrices can be proven analytically, using two 3D right-handed Cartesian coordinate systems with common origin and different orientations. This proof was tested using MATLAB to show a specific sequence of transformations applied to a Cartesian coordinate system. Using the results of this code, this report investigates the fact that the position of each transformation, relative to the original system, will have a different direction based on the sequences of the transformations, as we can see in the figures throughout. The results of Part B of this project involve discussion of the different components of the AP coordinate system, and the process of determining the coordinates of a specific star. For Part C of this project, a major aspect of revolved around a formal understanding of coordinate systems. This report will discuss the various ways to represent a specific location in many different reference systems (LA, AP, CT….). Through the understanding of this coordinate system, computer codes were created to solve problems involving the movements of the moon, and ecliptic latitude and longitude.

Table of Contents

Introduction/Objectives

In the first project for Geodetic Concepts, students were introduced to the use of Cartesian coordinate system transformations.

Part A

Part A of this project was a brief review of advanced calculus, and included a question about working with the rotations and reflections of a 3D right-handed Cartesian coordinate system. Questions 1 and 2 of the lab involved completing proofs involving rotation matrices, while question 3 allowed students to learn about the importance of the sequence of a set of transformations on a Cartesian coordinate system.

Part B

Part B of this lab involved research of different components of the Apparent Place (AP) coordinate system. The AP system is the apparent right ascension system, and it is the following transformation from the Instantaneous Terrestrial (IT) coordinate system. The IT system is the natural system of coordinates and it is fixed to the Earth. The AP and IT systems change position within the Earth with time, because the spin axis moves with respect to the Earth. This is also known as polar motion. Through this research, students were asked to determine the time that the moon will rise on September 30, 2018, as seen from the York University Astronomical Observatory, and its Azimuth. The moon rise is when the upper half of the moon is above the horizon. Since the Moon revolves around the Earth, its position changes rapidly with respect to the Sun. Within 24 hours its position changes by about 12 degrees, in the counterclockwise direction. The Azimuth is the horizontal direction of the moon at moonrise and moonset. The Azimuth can range from 0-360 degrees, measured in a clockwise direction. North has a value of 0 degrees, East 90 degrees, South 180 degrees, and West 270 degrees. Students were also asked to determine what time the Moon will cross the local meridian, taking into consideration the phase of the moon on that day.

Part C

Part C of this assignment requests a sophisticated software package, written in any scientific programming language, which can transform coordinates from the Local Astronomical (LA) system to the Ecliptic (EL) system and vice versa, with the option included to choose any two intermediate coordinate systems. A flowchart of the code structure is required. Using this software package, considering the York University Astronomical Observatory as the origin of the instantaneous LA system, calculate the transformed coordinates of a star in the AP system, calculate the coordinates of the observatory in the CT system, calculate the Azimuth and Zenith of the observatory, and convert the aforementioned star to Ecliptic coordinates.

Description of Software Structure

Part A

Question 1

The code written for question one was simply a test to check the analytical proof of the two rotation matrices, R1 and R2. This analytical proof is included in the Processing Inputs section of this report. The two rotation matrices were hardcoded into Matlab, with theta as thirty degrees. Each of the rotation matrices were multiplied by a vector, to test the rotation. By running this code, two new vectors are created. These vectors represent the rotated vectors.

Question 2

The code written for question two was also a simple test to check the analytical proof, included in the Processing Inputs section of this report. The rotation matrix R1 from question one was used for this proof. R1 was hardcoded in Matlab, and a user input was used to allow any angle of theta to be entered. The inverse and transpose of R1 were found, followed by the result of using negative theta in the matrix. A function was created to find the inverse of the matrix, and this function is also included in the Appendix of Software for Part A. When the code is run, all three of these resulting matrices are equal.

Question 3

The code written for question three of this project, was created to compute and show graphically the position of a set of transformations to a coordinate system, relative to the original system. Each transformation matrix was hardcoded in Matlab. The vector used to define the original system was defined on the first line of the code. After definition of each transformation matrix, the multiplication for each transformation was coded. To show these transformations graphically, vectors were created from the origin to the new position of the original coordinate system. Each of these vectors were plotted in a different colour, for easy identification.

Part B

Question 1

The code written for question one of Part B, included iterations to find the appropriate GAST and right ascension values. This code started with storing the GAST and right ascension values from the Almanac. The tolerance for the iteration is then defined, and a nested for-loop was created to find the position in the matrices for GAST and right ascension values, there the most precise values are located.

Question 2

The code written for question two of Part B, included all the equations to find approximate values for the hour angle, altitude and azimuth. This was done by coding the relevant equations in MatLab, and plugging in all the given values. The input function at the beginning of the code was used to show that for each date there is a corresponding altitude and azimuth.

Part C

The code written for Part C was to be used to transform between coordinate systems. In this lab, the objective is to design and implement a software package that calculates coordinate transformations from the Local Astronomical (LA) system to the Elliptical (E) system and vice versa. The software package is written using MATLAB and consists of 56 possible transformations. There are 8 systems and a total of 64 possible transformations (8 x 8 = 64). However, the transformation will be between AP, LA, CT, etc. The code will provide the user to choose an initial and final system as well as the values for first coordinate system. Depends on the user's values, the code will further request values based off the given values or the one has been calculated. The code will run until the transformation is complete, so the system will solve itself and convert from system to another based off your code structure.

Methodology/Processing Inputs – Part A

Question 1

1) Proof for the rotation matrix about x-axis.

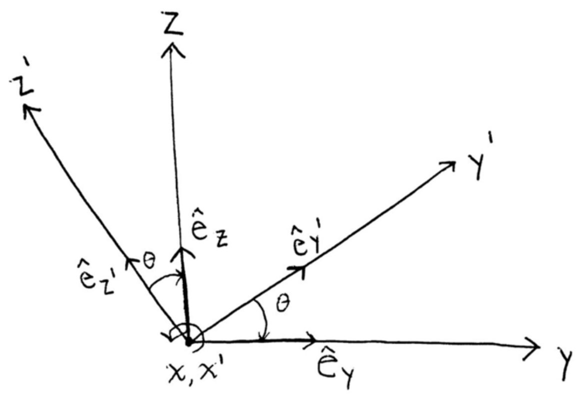


Figure 1:Rotation about the X-Axis on two 3D right-handed Cartesian coordinate systems Oxyz and Ox'y'z' with common origin and different orientation.

As seen in ***figure 1***, the Cartesian coordinate system is rotating about the x-axis. This means that the unit vector in the x direction, remains stationary. Therefore, the unit vectors in the x direction on both coordinate systems (Oxyz and Ox'y'z'), are equal:

$$\hat{e}_{x\prime} = \hat{e}_x$$

The y-axis and the z-axis are both rotated by an angle θ, as seen in ***figure 1.*** Using trigonometry, the position of the unit vectors for the z and y directions, in the Ox'y'z' system, can be found:

$$\hat{e}_{y\prime} = \hat{e}_y \cos\theta + \hat{e}_z \sin\theta$$
$$\hat{e}_{z\prime} = \hat{e}_y \cos(\theta + \tfrac{\pi}{2}) + \hat{e}_z \cos\theta = -\hat{e}_y \sin\theta + \hat{e}_z \cos\theta$$

Putting all the individual unit vectors together, gives the rotation matrix about the x-axis:

$$\begin{bmatrix} \hat{e}_{x'} \\ \hat{e}_{y'} \\ \hat{e}_{z'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{bmatrix}$$

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

2) Proof for the rotation matrix about the y-axis.

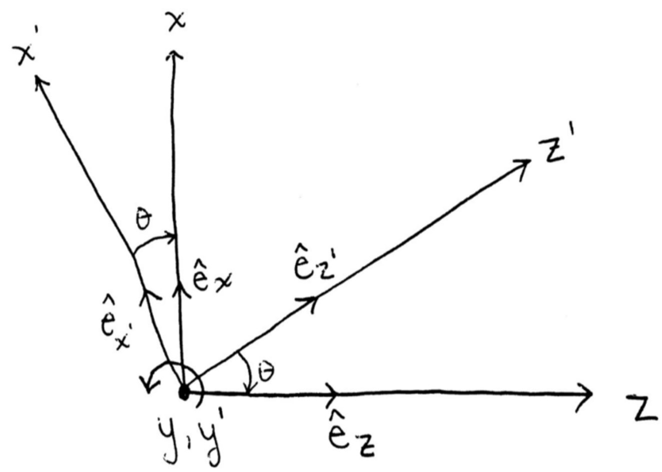


Figure 2: Rotation about the Y-Axis on two 3D right-handed Cartesian coordinate systems Oxyz and Ox'y'z' with common origin and different orientation.

As seen in ***figure 2***, the Cartesian coordinate system is rotating about the y-axis. This means that the unit vector in the y direction, remains stationary. Therefore, the unit vectors in the y direction on both coordinate systems (Oxyz and Ox'y'z'), are equal:

$$\hat{e}_{y'} = \hat{e}_y$$

The x-axis and the z-axis are both rotated by an angle θ, as seen in ***figure 2.*** Using trigonometry, the position of the unit vectors for the z and x directions, in the Ox'y'z' system, can be found:

$$\hat{e}_{z'} = \hat{e}_z \cos\theta + \hat{e}_x \sin\theta$$
$$\hat{e}_{x'} = \hat{e}_z \cos(\theta + \tfrac{\pi}{2}) + \hat{e}_x \cos\theta = -\hat{e}_z \sin\theta + \hat{e}_x \cos\theta$$

Putting all the individual unit vectors together, gives the rotation matrix about the y-axis:

$$\begin{bmatrix} \hat{e}_{x'} \\ \hat{e}_{y'} \\ \hat{e}_{z'} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{bmatrix}$$

$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

A sample code for question 1 is included in the Appendix of Software.

Question 2

Prove analytically that the following properties hold true: $\mathbf{R}^{-1}(q) = \mathbf{R}^{T}(q) = \mathbf{R}(-q)$.

Let the matrix $R(q) = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$

Trigonometric identities used in the problem:

(1) $\cos^2\theta + \sin^2\theta = 1$
(2) $\cos(-\theta) = cos\theta$
(3) $\sin(-\theta) = -sin\theta$

The first step in this proof is to show that the determinant of any rotation matrix is 1. This is due to the following matrix property:

$$R^{-1} = \frac{1}{\det{(R)}} R^T$$

Given the matrix $R(q)$ above, the determinant would be calculated as:

$$det(R) = a_1(b_2c_3 - b_3c_2) - a_2(b_1c_3 - b_3c_1) + a_3(b_1c_2 - b_2c_1)$$

Now using the matrix $R_1(\theta)$ from question 1:

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

$$det(R_1) = (\cos^2\theta + \sin^2\theta) - 0 + 0$$

Due to identity (1):

$$det(R_1) = 1$$

After finding the determinant of the rotation matrix, the next step is to find the transpose of the matrix. To find the transpose of the matrix, simply swap the rows and the columns:

$$R_1^T(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}^T$$

$$\boldsymbol{R_1^T(\theta)} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\cos\theta} & \boldsymbol{-\sin\theta} \\ \mathbf{0} & \boldsymbol{\sin\theta} & \boldsymbol{\cos\theta} \end{bmatrix}$$

Now, the inverse of the matrix can be found, using the matrix property from above:

$$R^{-1} = \frac{1}{\det(R)} R^T$$

$$R_1^{-1} = 1 * R_1^T$$

$$R_1^{-1} = R_1^T$$

$$\boldsymbol{R_1^{-1}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\cos\theta} & \boldsymbol{-\sin\theta} \\ \mathbf{0} & \boldsymbol{\sin\theta} & \boldsymbol{\cos\theta} \end{bmatrix}$$

Finally, to find matrix R(-*q*), plug in $-\theta$ anywhere there is a θ in the matrix:

$$R_1(-\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & \sin(-\theta) \\ 0 & -\sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

Due to identities (2) and (3):

$$\boldsymbol{R_1(-\theta)} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\cos(\theta)} & \boldsymbol{-\sin(\theta)} \\ \mathbf{0} & \boldsymbol{\sin(\theta)} & \boldsymbol{\cos(\theta)} \end{bmatrix}$$

Therefore, $\boldsymbol{R_1^{-1}(\theta)} = \boldsymbol{R_1^T(\theta)} = \boldsymbol{R_1(-\theta)}$

And in general: **$\mathbf{R}^{-1}(q) = \mathbf{R}^{T}(q) = \mathbf{R}(-q)$.**

A sample code for question 2 is included in the Appendix of Software.

Processing Outputs – Part A

Question 1

A sample code for question 1 is included in the Appendix of Software.

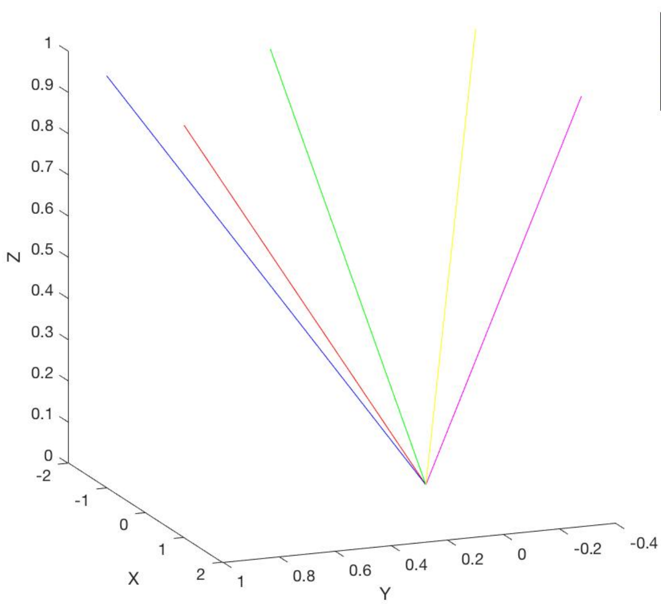
Question 2

A sample code for question 2 is included in the Appendix of Software.

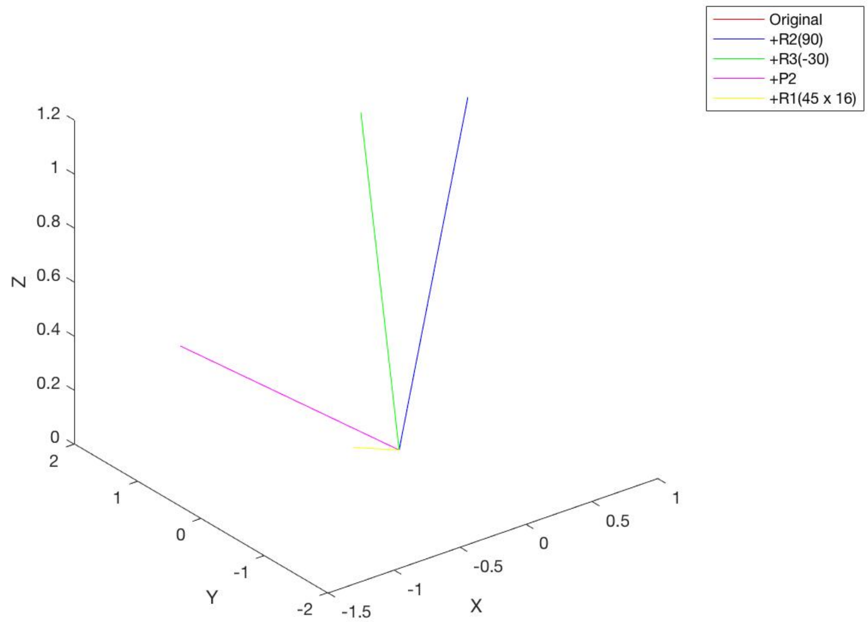
Question 3

See the **Appendix of Software** for fully commented code, and see the **Appendix of Figures** for more figures of the plot with corresponding descriptions.

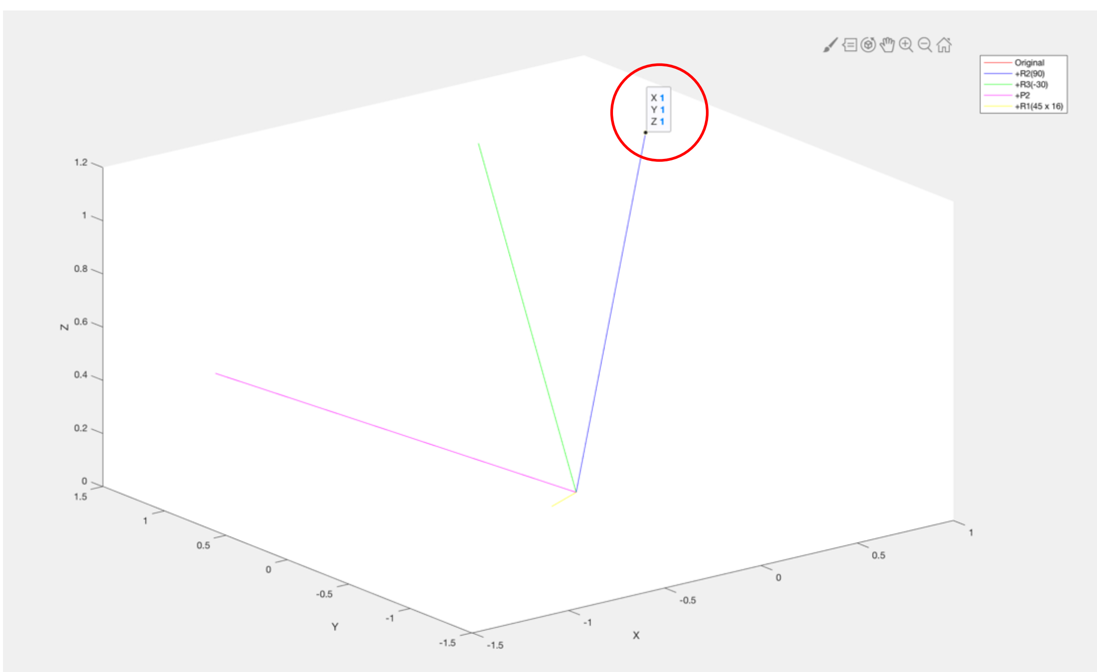
Transformation Order: $R_2(90)$, $R_3(-30)$, P_2, $R_1(45x16)$



Different Transformation Order: $R_1(45x16)$, $R_3(-30)$, $R_2(90)$, P_2



With this order of transformations, the original vector is overlapping the $R_2(90)$ transformation line in blue. The coordinates of the terminal point can be seen in the figure below.



Methodology/Processing Inputs – Part B

Question 1

Question one asked students to determine what time the Moon will rise on September 30, 2018, as seen from York University Astronomical Observatory. It also asked students to determine the Moon's azimuth, and the time it will cross the local meridian. The given values include the Astronomical Latitude and Longitude of the York Observatory:

Latitude: $\phi = 43°46'30"N$
Longitude: $\lambda = 280°11'06"E$

To complete this question, the situation must be understood. To determine the time that the Moon will rise, based on the observer's meridian, the AP coordinate system must be understood. Figure 1 shows this coordinate system and its components.

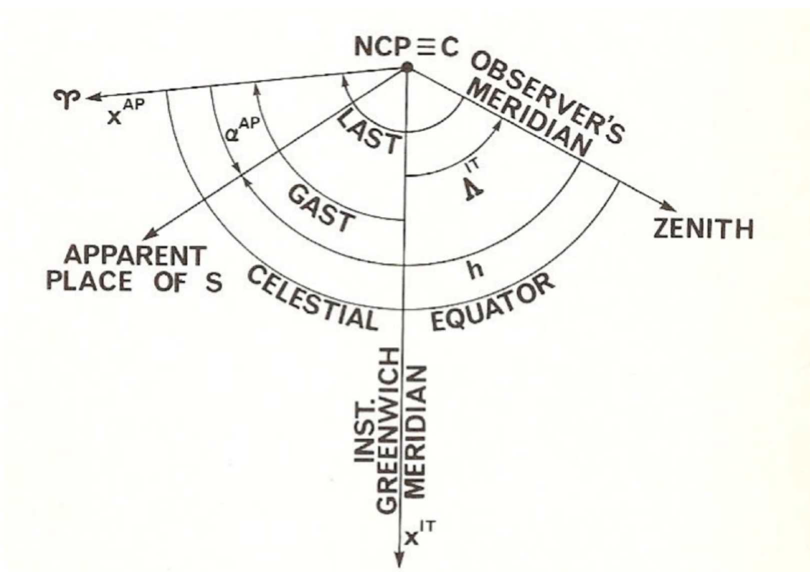


Figure 1: The Apparent Place Coordinate System [2].

- The **Observer's Meridian** represents the direction of the eye line of the observer.
- **LAST** is the Local Apparent Sidereal Time
- $\boldsymbol{\lambda_{IT}}$ is the Astronomical Longitude in the IT-system.
- $\boldsymbol{h}$ is the hour angle
- **GAST** is Greenwich Apparent Sidereal Time
- $\boldsymbol{\alpha_{AP}}$ is the Right Ascension in the AP-system.
- **The Apparent Place of S** represents the location of the Moon
- $\boldsymbol{ZENITH}$ is the imaginary point exactly above the particular position.

- $\boldsymbol{INST.Greenwich\ Meridian\ X^{IT}}$ – Axis Moves with the time, an imaginary line used to indicate 0° longitude.

To calculate the time that the Moon will rise, the Apparent Place of S must lie on the Observer's Meridian. Therefore, the hour angle must be zero. For this to be true, the following equation must be satisfied:

$$h = GAST + \lambda - \alpha_{AP}$$
$$0 = GAST + \lambda - \alpha_{AP}$$

The GAST and right ascension angles were found in the "*Astronomical Ephemeris and Nautical Almanac*". Based on the given astronomical longitude, the following values were found from this Almanac:

	September 30, 2018	**October 1, 2018**
GAST (h m s)	0 34 45.9328	0 38 42.4897
α_{AP} (h m s)	4 18 22.74	5 16 0.44

Both the GAST and the right ascension values were stored in separate matrices, with increments by the seconds. This ensured the most precise values would be used in the next iteration.

An interpolation had to be coded to satisfy the above equation: $0 = GAST + \lambda - \alpha_{AP}$. A code was written to iterate through all GAST and right ascension angles to find the two values which best approximated for h = 0.

Question 2

For question two, students started by finding the Julian Date of September 30th (using an online converter) and then plugged it in the year past J2000 to get the T(JD). The second step, involved getting the values $\Delta\alpha = T(JD) * \alpha$ and $\Delta\delta = T(JD) * \delta$. The values for alpha and the declination were changed to decimal hours. The coordinates in the hour angle system were found using the following equation:

$$h = GAST + \lambda - \alpha_{true}$$

However, the value for GAST had to be calculated using the following equation:

GAST = 0.00 UT + Group#

And the value for $\alpha_{true} = \alpha + \Delta\alpha$.

For question c, two equations were used to obtain the values for the altitude and the azimuth:

$$sin\phi sin\delta + cos\phi cos\delta cosh - cosZ = 0$$

Where Z is the altitude.

$$\tan A = \frac{sinh}{sin\phi cosh - tan\delta cos\phi}$$

Where A is the azimuth.

Processing Outputs – Part B

Question 1

Through the MatLab interpolation, the value of GAST that was found was 0.579425777777778 decimal hours, and the value of the right ascension angle was 4.742 decimal hours. The hour angle was equal to approximately -8.9e-7 using these values, which is extremely close to zero. From this GAST value, the calculated moon rise time was found to be the GAST time (00:34:36) minus 4 hours. To convert to local time, 4 hours was subtracted. The results of question one can be found in ***Table 1*** below.

Time of Moon Rise	20:34:36 or 8:34:36 pm
Azimuth	

What time will the Moon rise on 30 September 2018, as seen from York University Astronomical Observatory (F = 43°46’30”N and L = 280°11’06”E)? Determine its **azimuth**. What time will the Moon cross the local meridian? Please explain the results, taking into consideration the phase of the moon that day. Your answers must be in local time. [20 points]

Question 2

Question two of Part B yielded the following results, included in ***Table 2***:

September 30	Values
h	1.8386e4 hours
Azimuth	1.7890 degrees
Altitude	0.4799 degrees

Table 2: Results of question 2.

Methodology/Processing Inputs – Part C

In Part C, students were asked to find a way to translate from one coordinate system to another, using intermediate systems. Below is a brief overview of each coordinate system as well as their respective transformations to other systems.

Local Astronomical System

- The system in which stars are observed.
- Conventional Earth-Spin Axis and the gravity vector of the observer define this system.
- Geotheodolite and theodolite plume line sense the directions, respectively.

Instantaneous Terrestrial System

- Natural system of coordinates
- Geocentric; fixed to earth.

Apparent Place (Apparent Right Ascension) System

- Uses the centre of mass on earth as an origin.

Conventional Terrestrial System

- Used as the closest possible approximation of the IT system, in a new coordinate system solidly attached to earth.
- Likely the most important terrestrial system out of them all.

True Right Ascension System

- The most important celestial system of them all.
- Heliocentric; fixed to the sun.
- Sometimes used to catalogue star positions.

Mean Right Ascension System

- The coordinate system used in star catalogues.
- Separated into MRA(τ) and MRA(τ_0), which differ via the effect of precession.

Ecliptic System

- An approximation of an inertial system, which is motionless in respect to galaxies.
- Heliocentric.
- Star catalogues provide coordinates in respect to the Ecliptic system.

Local Astronomical to Instantaneous Terrestrial System

$r^{IT} = R_3(\pi-\Lambda)*R_2(\pi/2 - \Phi)r^{LA}$ to $r^{LA} = R_2^T(\pi/2- \Phi)*R_3^T(\pi/2 -\Lambda)r^{IT}$

Apparent Place System to Instantaneous Terrestrial

$r^{AP} = R_3(-GAST^T r^{IT}$ to $r^{IT} = R_3(GAST)^T r^{AP}$

Instantaneous Terrestrial from Conventional Terrestrial

$r^{IT} = R_1(Y_p)R_2(X_p)r^{CT}$

True Right Ascension from Mean Right Ascension

$r^{TRA} = R_1(-\varepsilon-\Delta\varepsilon)R_3(\Psi)R_1(\varepsilon)r^{MRA(}\tau)$

Mean Right Ascension(τ) to Mean Right Ascension(τ)

$r^{MRA(}\tau) = R_3(-z)R_2(\theta)R_3(-\varsigma_0)r^{MRA(}\tau_0)$

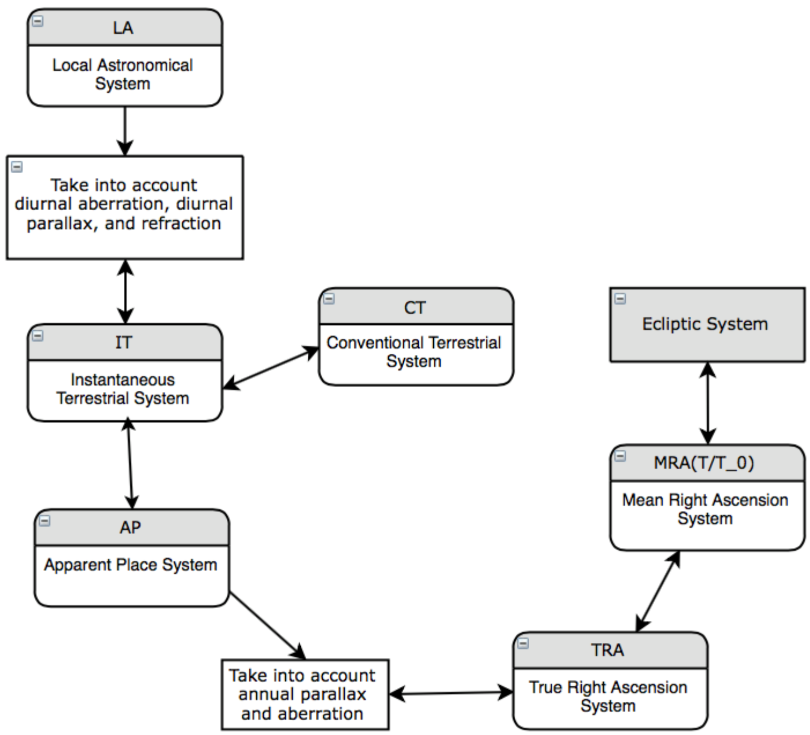
Mean Right Ascension(τ_0) to Mean Right Ascension(τ)

$$r^{MRA}(\tau) = R_3(-z)R_2(\theta)R_3(-\varsigma_0)r^{MRA}(\tau_0)$$

Mean Right Ascension from Ecliptic

$$r^{MRA} = R_1(\varepsilon)r^{-E}$$

Flow Chart of Code Delivery



Processing Outputs – Part C

Question C – AP to LA

$\alpha = 2h\ 50m\ 53.97s$ =42.72°

$\delta = 89°19'14.98'' = 89.32082778$

$\phi = 43°46'30'' = 43.775°$ - Based on Question B – might change

$\lambda = 280°11'06'' = 280.185°$ - Based on Question B – might change

Time = 04:00 UT Oct 1st 2018

GAST = 04°39'21.9161''

$h = GAST - \alpha + \lambda = 242.1°$

Julian Date: 2458392.666667

Using the code:

Your LA coordinates are:
-0.337629074550727
-0.732016240952803
0.722644701208887

Your is latitude and longitude are:
77.706455554794 -72.3242992011233

Your IT coordinates are:
-0.760084548397185
-0.268537982953066
0.722644701208887

Your latitude and longitude are:
46.2732657661715 -160.541677963925

>>

To find the Zenith

$$\cos(z) = \sin(\phi)\sin(\delta) + \cos(\phi)\cos(\delta)\cos(h)$$

$$z = \cos^{-1}(\sin(43.775^\circ)\sin(89.32082778^\circ) + \cos(43.775^\circ)\sin(89.32082778^\circ)\cos(271.14^\circ)) = 46.50^\circ$$

To find the Azimuth

$$\tan(A) = \frac{\sin(h)}{\sin(\phi)\sin(h) - \tan(\delta)\cos(\phi)}$$

$$= \tan^{-1}\frac{\sin(271.14^\circ)}{\sin(43.775^\circ)\sin(271.14^\circ) - \tan(89.32082778^\circ)\cos(43.775^\circ)} = 0.80^\circ$$

Question A

Sample Data Processing, Analysis and Discussion

Part A Discussion

After the completion of Part A of this project, students have expanded their knowledge of the processes involved with Cartesian Coordinate Transformations. Questions 1 and 2 were a review of some advanced calculus matrix techniques, and helped aid students in the visualization of the transformations, themselves. Through the work completed on Part A of the project, the group has gained a better understanding of these topics.

Question one of Part A involved deriving the rotation matrices R1(q) and R2(q) about axes X and Y, respectively. The most important step in this question was the visualization of the right-handed coordinate system. By using the right-hand rule, the group was able to see which directions each axis should point. In the beginning, there was confusion about how to define a right-handed coordinate system, as opposed to a left-handed coordinate system, but once this was resolved the problem only took a matter of minutes to complete. After the initial visualization of both of the Cartesian coordinate systems (i.e. the original coordinate system and the rotated coordinate system), the next step was to use trigonometry to define the rotations of the unit vectors. Once this was finished, each vector was put together to form the rotation matrix around the x-axis and the same for around the y-axis. A code was written to test how the completed proof worked.

Question two of Part A introduced students to the properties of rotation matrices. The group decided to use the matrix R_1 from question 1 to explain these properties in detail. A program was written for this question, to prove that the proof stands for all values of theta. A user input was used. Through the understanding of how these properties work, the students became more familiar with how rotation matrices work.

Question three of Part A demonstrated that the order of transformations applied to a plot or a vector is very important. The completion of these transformations in any other order would present a different result. However, no matter what order the transformations are applied, the combination of these transformations can only be pre-multiplied by the vector A, not vice-versa. This is because these matrices are not commutative. The code begins with defining the original coordinate system, the rotation about the x-axis, the rotation about the y-axis, the rotation about the z-axis, and the reflection across the y-axis. From there, the original vector was multiplied by the transformation matrices in one order, with each transformation shown in the plot. Conversely, there is an example of the same transformations done in a different order, giving a different resulting plot.

Part B Discussion

Question one of Part B involved gaining a better understanding of the Apparent Place coordinate system. Through the understanding of this system, iterations were coded in Matlab to provide the team with the most accurate values of GAST and right ascension angles in order to find the exact time that the moon would rise on September 30, 2018. Moon rise is when the upper part of the moon is above the horizon, and sets when the lower part is about to disappear below the horizon. Since the Moon revolves around the Earth, its position changes rapidly with respect to the Sun, as mentioned in the Introduction. This question also asked students to find the Azimuth of the Moon. The Azimuth is the horizontal direction of moon at moonrise and moonset. Based on the acquired knowledge of this coordinate system, question one included both written code, and research through the Astronomical Almanac. Based on the necessary calculations, the Moon will rise on 30th September 2018 at (-----) with azimuth angle of (----) degrees. The moon will cross the local meridian at (----) with illumination of (----) on that day.

Question two of Part B involved working with equations to find approximate values for an hour angle, altitude, and azimuth of the star Persei for September 30, 2018 at midnight. Using the given values for right ascension, apparent declination, and proper motions for both values, the coordinate of the star was found with respect to York University's Astronomical Observatory. Students also were asked to find the horizon system coordinates of the star at the observatory.

Part C Discussion

Part C serves to enhance the students' understanding of the various coordinate systems used in Geodesy and how they are interconnected via a series of transformations. Preceding the main questions of the Part, a software package that has the ability to receive user input and determine the appropriate transformations necessary to achieve coordinates in the requested output system was requested. This software package forms the backbone of the work necessary to complete the exercise, with the exercise itself largely acting as a check of the software's capabilities. As such, the first question of Part C requests a simple conversion of LA coordinates into the AP system. This is easily found using the software package, simply plugging in the given coordinates and the appropriate Julian Date for the according time will produce accurately-transformed coordinates in the AP system. The second question requires use of an Almanac to determine the polar motion parameters Xp and Yp. After obtaining these values, they are used to convert IT coordinates to CT coordinates. The third and final question for this exercise requests a set of LA coordinates (Zenith and Azimuth), with given values being AP system coordinates. Using the software package, plugging in all the relevant given values and using a different Julian Date as specified in the exercise, will yield an accurate set of LA system coordinates.

Division of Labor

Victoria Donko	Talon Denton	Yaseen Al-Taie	Dion Farquhar
- <u>Report:</u> Introduction, Description of Software, part of Discussion part A and B, Summary, created Appendices. - Analytical proof and coded questions 1 and 2 for Part A - Edited code in question 3 Part A - Coded question 1 for part B	- <u>Report:</u> Part of Discussion - Coded question 3 for part A - Question 2-part B and C (the code and the report) - Discussion, analytical proof for question 1	- <u>Report:</u> Part of report Discussion, Executive Summary, analytical proof for question 1. - Experimented with iterations in Excel for Part B question 1 - Question 2-part B and C (part of the code and the report)	- <u>Report:</u> Summary for Part C, methodology for Part C - Coding Part C

Summary

Part A

Part A of this project included a reminder of the application of matrix properties, as well experimentation with how to code these types of problems. Through the completion of Part A, students have gained a new understanding of rotation and transformation matrices, and this will prove useful in other assignments in Geodetic Concepts, to follow.

Part B

Part B of this project, was a complete introduction to the Apparent Place (AP) coordinate system. Through completing question 1 of this part of the project, students gained an understanding of the fundamental components of this coordinate system. These included the Greenwich Apparent Sidereal Time (GAST) angle, and the Right Ascension angle. The coding of Part B was more difficult than Part A, as these concepts were brand new to students. Through completing iterations in Matlab, students could understand more about how the AP coordinate system operates.

Part C

Part C of this project was an introduction to multiple coordinate systems, and the concept of coordinate transformations. By writing a comprehensive software package that can correctly accept user input and perform transformations from the system of origin to the system of choice via intermediate systems, students have gained and demonstrated knowledge of how coordinate systems work, differ, and represent things differently.

References

[1] Bisnath, Sunil. (2018). LE/ESSE 3610 Geodetic Concepts: Lecture Slides [PowerPoint Slides].

[2] P. Vanicek and E. Krakiwsky, *Geodesy: The Concepts.* Amsterdam: Elsevier, 1992.

http://aa.usno.navy.mil/data/docs/JulianDate.php

Appendix of Figures

Figure 1A: Question 3 Cartesian Coordinate System Transformations.

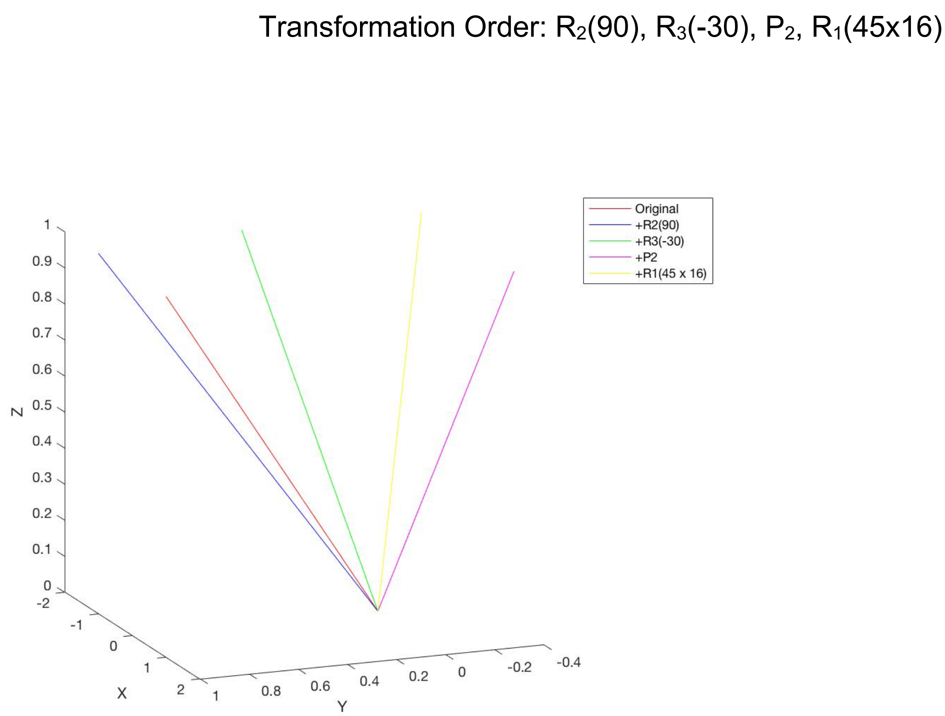


Figure 2A: Question 3, Y-Z Axis.

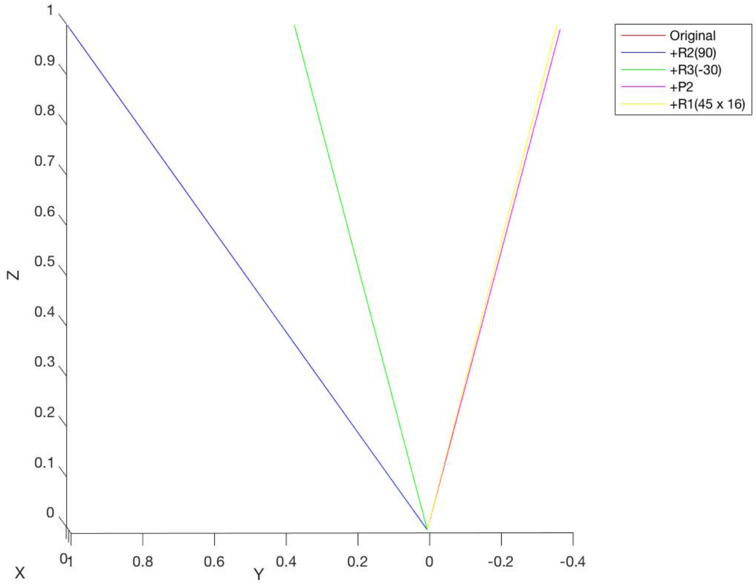


Figure 3A: Question 3, Z-X Axis.

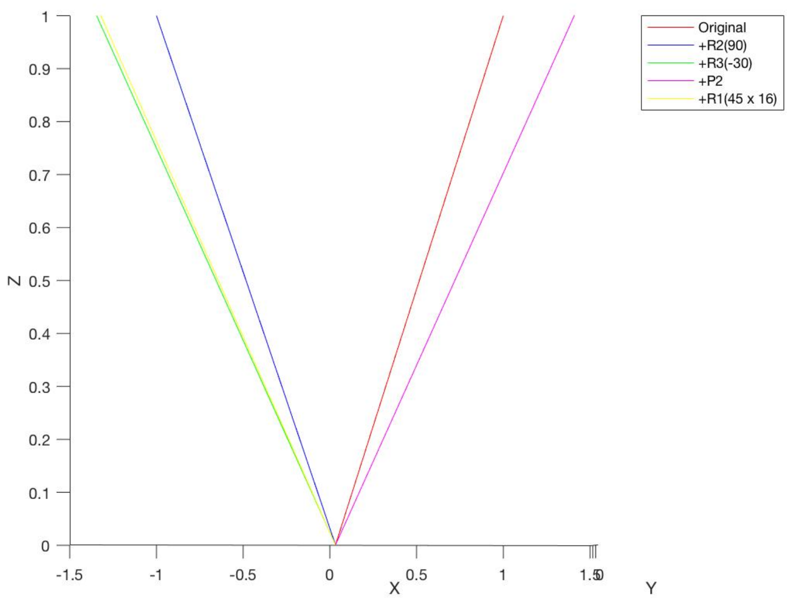


Figure 4A: Question 3, Y-X Axis.

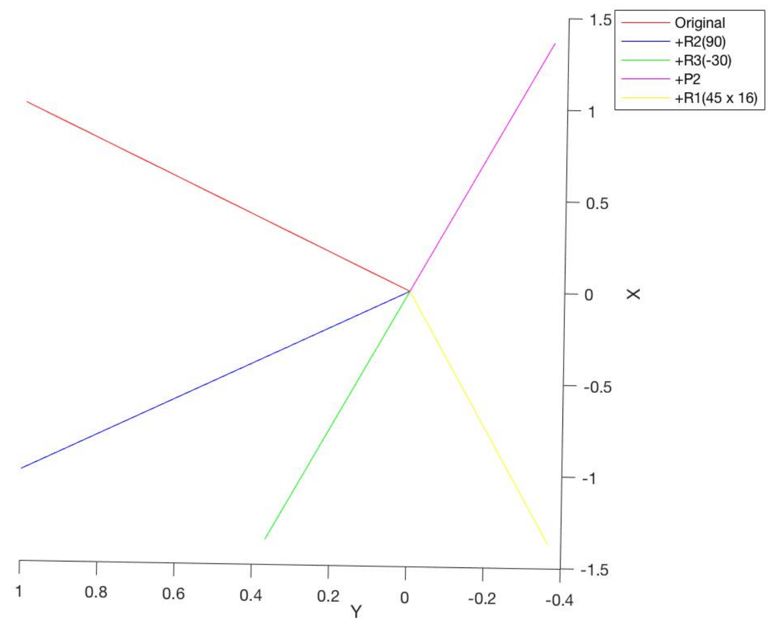


Figure 5A: Question 3 Cartesian Coordinate System – Z-X Axis with a different order of transformations.

Transformation Order: $R_1(45x16)$, $R_3(-30)$, $R_2(90)$, P_2

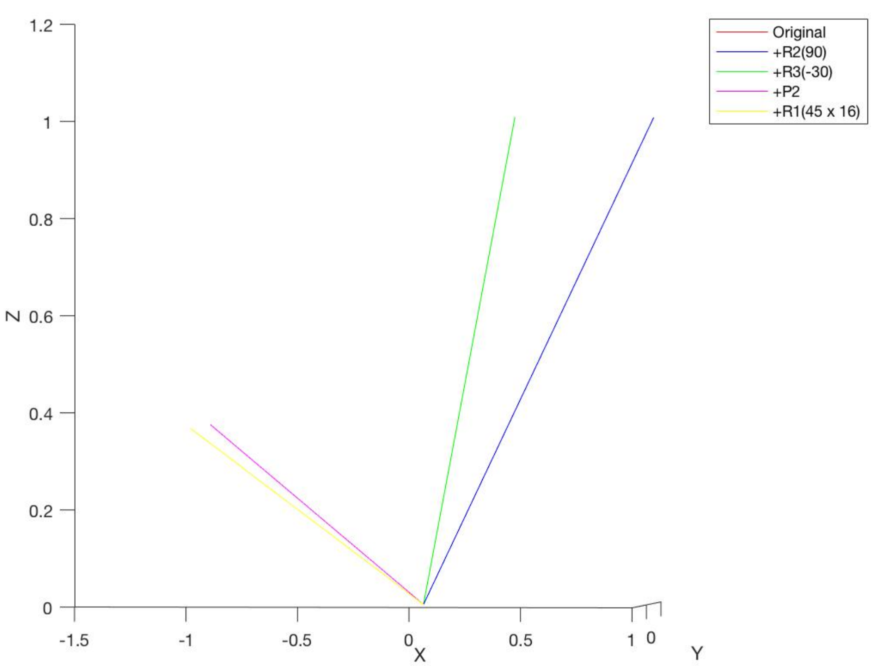
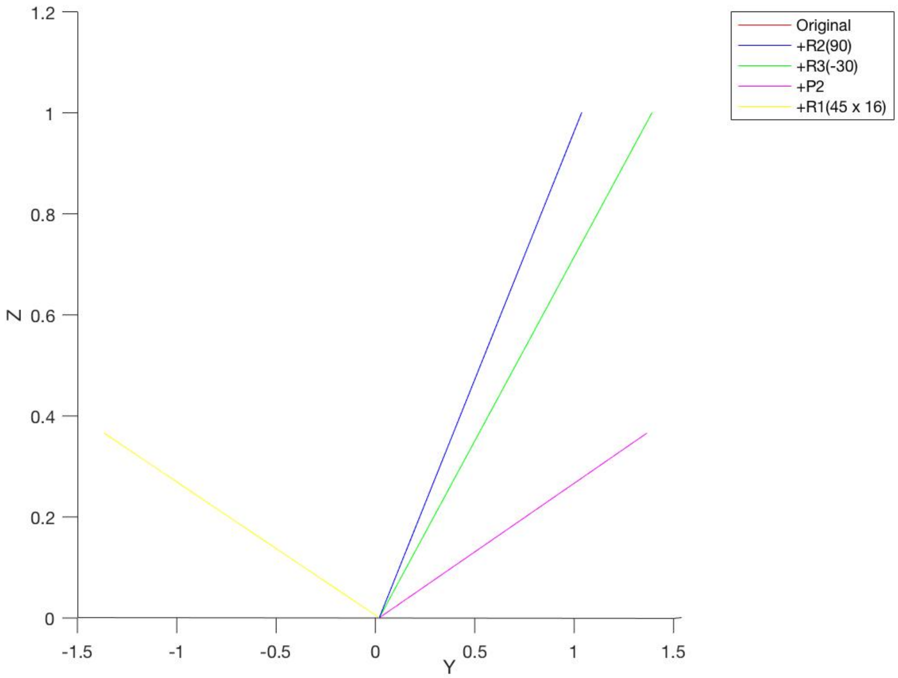


Figure 6A: Question 3, Z-Y axis with a different order of transformations.



Part C - Question A

```
% ESSE-3610 Geodetic Concepts Lab 2
function Geodetics_Project1
    disp('These are the list of transformations:');
    disp('0 - CT/LA');
    disp('1 - IT');
    disp('2 - AP');
    disp('3 - TRA');
    disp('4 - MRA(T)');
    disp('5 - MRA(T_0)');
    disp('6 - Ecliptic');
    A = input('Please enter start: ');
    B = input('Please enter stop: ');

    if (A > 6) || (A < 0) || (B > 6) || (B < 0)
        disp('Incorrect values, please run this function again.');
    else
        if A == 0 || B == 0
            an = input('0 - CT or 1 - LA? ');
            if an == 0
                if (A < B)
                    positive(A, B);
                else
                    negative(B, A);
                end
            elseif an == 1
                if (A < B)
                    A = A + 1;
                    positive(A, B);
                else
                    B = B + 1;
                    negative(B, A);
                end
```

```
            end
        else
            if (A < B)
                A = A + 1;
                positive(A, B);
            else
                B = B + 1;
                negative(B, A);
            end
        end
    end
end

function positive(A, B)
    phi = input('What is your altitude value? ');
    lambda = input('What is your azimuth value? ');
    jd = input('What is your Julian Date? ');
    C = 0;
    val = [phi, lambda, jd, cosd(phi)*cosd(lambda), cosd(phi)*sind(lambda), sind(phi)];
    disp('These are your starting values: ');
    disp([val(end-2) val(end-1) val(end)]);
    for A = A:B
        if A == 0
           x_p = input('What is x_p: ');
           y_p = input('What is y_p: ');
           m = CT_to_IT(x_p, y_p);
           ct = [val(end-2); val(end -1); val(end)];
           it = (m*ct)';
           val(end+1) = it(end-2); %#ok<*AGROW>
           val(end+1) = it(end-1);
           val(end+1) = it(end);
           disp('Your IT coordinates are: ');
           disp(it);
           disp('Your new latitude and longitude are: ');
           [phi, lambda] = xyzToAngles(it(end-2), it(end-1), it(end));
           disp([phi, lambda]);
           C = 1;

        elseif A == 1
           if C == 0
               phi = input('What is the latitude: ');
               lambda = input('What is the longitude: ');
               r = LA_TO_IT(phi,lambda);
               la = [val(end-2); val(end -1); val(end)];
               it = (r*la)';
               val(end+1) = it(end-2);
               val(end+1) = it(end-1);
               val(end+1) = it(end);
               disp('Your IT coordinates are: ');
               disp(it);
               disp('Your new latitude and longitude are: ');
               [phi, lambda] = xyzToAngles(it(end-2), it(end-1), it(end));
               disp([phi, lambda]);
           end

        elseif A == 2
           g = IT_TO_AP(jd);
           it = [val(end-2); val(end -1); val(end)];
           ap = (g*it)';
           val(end+1) = ap(end-2);
           val(end+1) = ap(end-1);
           val(end+1) = ap(end);
           disp('Your AP coordinates are: ');
           disp(ap);
           disp('Your declination and right ascension are: ');
           [delta, alpha] = xyzToAngles(it(end-2), it(end-1), it(end));
           disp([delta, alpha]);

        elseif A == 3
           ap = [val(end-2); val(end -1); val(end)];
           [delta, alpha] = xyzToAngles(ap(end-2), ap(end-1), ap(end));
           r = input('What is the distance from the Sun to the object? ');
           [delta_ra, delta_dec] = parallax(ra,dec);
           delta = delta + delta_dec;
           alpha = alpha + delta_ra;
           x = cos(delta)*cos(alpha);
           y = cos(delta)*sin(alpha);
           z = sin(delta);
```

```
            [delta_a, delta_d] = AP_to_TRA(alpha, delta, x, y, z);
            delta = delta + delta_d;
            alpha = alpha + delta_a;
            x = cos(delta)*cos(alpha);
            y = cos(delta)*sin(alpha);
            z = sin(delta);
            val(end+1) = x;
            val(end+1) = y;
            val(end+1) = z;
            tra = [x y z];
            disp('Your TRA coordinates are: ');
            disp(tra);
            disp('Your declination and right ascension are: ');
            disp([delta, alpha]);

         elseif A == 4
            tra = [val(end-2); val(end -1); val(end)];
            n = TRA_TO_MRAT(jd);
            mra = (n*tra)';
            val(end+1) = mra(end-2);
            val(end+1) = mra(end-1);
            val(end+1) = mra(end);
            disp('Your MRA(T) coordinates are: ');
            disp(mra);
            disp('Your declination and right ascension are: ');
            [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
            disp([delta, alpha]);

         elseif A == 5
            mra = [val(end-2); val(end -1); val(end)];
            p = MRAT_TO_MRATO(jd);
            mra_t0 = (p*mra)';
            val(end+1) = mra_t0(end-2);
            val(end+1) = mra_t0(end-1);
            val(end+1) = mra_t0(end);
            disp('Your MRA(T_0) coordinates are: ');
            disp(mra_t0);
            disp('Your declination and right ascension are: ');
            [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
            disp([delta, alpha]);

         elseif A == 6
            mra_t0 = [val(end-2); val(end -1); val(end)];
            o = MRATO_TO_E(jd);
            ec = (o*mra_t0)';
            val(end+1) = ec(end-2);
            val(end+1) = ec(end-1);
            val(end+1) = ec(end);
            disp('Your Ecliptic coordinates are: ');
            disp(mra_t0);
            disp('Your declination and right ascension are: ');
            [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
            disp([delta, alpha]);
         end
     end
end

function negative(A, B)
    phi = input('What is your latitude/declination/altitude value? ');
    lambda = input('What is your longitude/right ascension/azimuth value? ');
    jd = input('What is your Julian Date? ');
    val = [phi, lambda, jd, cos(phi)*cos(lambda), cos(phi)*sin(lambda), sin(phi)];
    C = 0;
    for A = A:B
        if B == 0
           x_p = input('What is x_p: ');
           y_p = input('What is y_p: ');
           m = CT_TO_IT(x_p, y_p);
           it = [val(end-2); val(end -1); val(end)];
           ct = (it/m)';
           val(end+1) = it(end-2);
           val(end+1) = it(end-1);
           val(end+1) = it(end);
           disp('Your CT coordinates are: ');
           disp(ct);
           disp('Your new phi and lambda are: ');
           [phi, lambda] = xyzToAngles(it(end-2), it(end-1), it(end));
```

```
        disp([phi, lambda]);
        C = 1;

    elseif A == 1
        if C == 0
            phi = input('What is the latitude: ');
            lambda = input('What is the longitude: ');
            r = LA_TO_IT(phi, lambda);
            it = [val(end-2); val(end -1); val(end)];
            la = (it/r)';
            val(end+1) = la(end-2);
            val(end+1) = la(end-1);
            val(end+1) = la(end);
            disp('Your LA coordinates are: ');
            disp(la);
            disp('Your is latitude and longitude are: ');
            [phi, lambda] = xyzToAngles(it(end-2), it(end-1), it(end));
            disp([phi, lambda]);
        end

    elseif A == 2
        g = IT_TO_AP(jd);
        ap = [val(end-2); val(end -1); val(end)];
        it = (ap/g)';
        val(end+1) = it(end-2);
        val(end+1) = it(end-1);
        val(end+1) = it(end);
        disp('Your IT coordinates are: ');
        disp(ct);
        disp('Your latitude and longitude are: ');
        [phi, lambda] = xyzToAngles(it(end-2), it(end-1), it(end));
        disp([phi, lambda]);

    elseif A == 3
        tra = [val(end-2); val(end -1); val(end)];
        [delta, alpha] = xyzToAngles(ap(end-2), ap(end-1), ap(end));
        r = input('What is the distance from the Sun to the object? ');
        [delta_ra, delta_dec] = parallax(alpha, delta, r, ap(end-2), ap(end-1), ap(end));
        delta = delta - delta_dec;
        alpha = alpha - delta_ra;
        x = cos(delta)*cos(alpha);
        y = cos(delta)*sin(alpha);
        z = sin(delta);

        [delta_a, delta_d] = AP_TO_TRA(alpha, delta, x, y, z);
        delta = delta - delta_a;
        alpha = alpha - delta_d;
        x = cos(delta)*cos(alpha);
        y = cos(delta)*sin(alpha);
        z = sin(delta);
        val(end+1) = x;
        val(end+1) = y;
        val(end+1) = z;
        ap = [x y z];
        disp('Your AP coordinates are: ');
        disp(ap);
        disp('Your declination and right ascension are: ');
        disp([delta, alpha]);

    elseif A == 4
        mra = [val(end-2); val(end -1); val(end)];
        n = TRA_TO_MRAT(jd);
        tra = (tra/n)';
        val(end+1) = tra(end-2);
        val(end+1) = tra(end-1);
        val(end+1) = tra(end);
        disp('Your TRA coordinates are: ');
        disp(tra);
        disp('Your declination and right ascension are: ');
        [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
        disp([delta, alpha]);

    elseif A == 5
        mra_t0 = [val(end-2); val(end -1); val(end)];
        p = MRAT_TO_MRATO(jd);
        mra = (mra_t0/p)';
        val(end+1) = mra(end-2);
        val(end+1) = mra(end-1);
```

```
            val(end+1) = mra(end);
            disp('Your MRA(T) coordinates are: ');
            disp(mra);
            disp('Your declination and right ascension are: ');
            [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
            disp([delta, alpha]);

        elseif A == 6
            ec = [val(end-2); val(end -1); val(end)];
            o = MRATO_TO_E(jd);
            mra_t0 = (ec/o)';
            val(end+1) = mra_t0(end-2);
            val(end+1) = mra_t0(end-1);
            val(end+1) = mra_t0(end);
            disp('Your MRA(T_0) coordinates are: ');
            disp(mra_t0);
            disp('Your declination and right ascension are: ');
            [delta, alpha] = xyzToAngles(mra(end-2), mra(end-1), mra(end));
            disp([delta, alpha]);
        end
    end
end
```

Function for IT to AP Systems

```
function g = IT_TO_AP(julian_date)
    d = julian_date - 2451545;
    gmst = 18.697374558+24.065570982441908*d;
    omega = 125.04-0.052954*d;
    L = 280.47+0.98565*d;
    delta_psi = 0.00319*sind(omega) - 0.000024*sind(2*L);
    epsilon = 23.4393-0.0000004*d;
    gast = (gmst + delta_psi*cosd(epsilon))/3600;
    g = [cosd(-gast) -sind(-gast) 0; sind(-gast) cosd(-gast) 0; 0 0 1];
    return
end
```