Project 2: Distance to Mars

Due: Thursday May 24, 11:59 pm

1 Assignment Overview

This project focuses on writing functions that do mathematical computations. You will compute the distance from Earth to Mars.

2 Assignment Deliverables

You must turn in completed versions of the following files:

- proj02/mars.cpp

Be sure to use the specified file name and to submit your files for grading via **Mimir** before the project deadline.

3 Background

In project 1 you computed the location of the Earth about Sol. For this project you will extend this to other planets and then compute the distance between them. Normally you can compute the distance d using the formula $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. However, these coordinates are in the coordinate spaces of their respective orbits. To compute the distance we will have to translate the coordinates of one of the planets from one system to another. If the two orbits of the two planets are out of phase at angle ϕ this can be done by the equations $x' = x \cos\phi - y \sin\phi$ and $y' = x \sin\phi + y \cos\phi$.

4 Assignment Specifications

In this and in future projects we will provide *exactly* our function specifications: the function name, its return type, its arguments, and each arguments type. The functions will be tested individually in Mimir using these exact function specifications. If you do not follow the function specifications, these independent tests of your functions will fail. Do not change the function declarations!

What you test on Mimir is a file that contains only the functions. You do not turn in a main program. We can test the functions individually on Mimir. However, you should write your own main program to test your functions separate from Mimir. It is more flexible and you can debug more easily.

Note that `Distance` is overloaded.

function `Eccentricity`
params: semi-major axis: double, semi-minor axis: double
returns: double

function `EccentricAnomaly`
params: eccentricity : double, true anomaly : double
returns: double

function `XPos`

params: semi-major axis : double, semi-minor axis : double, true anomaly : double

returns: double

function `YPos`

params: semi-major axis : double, semi-minor axis : double, true anomaly : double

returns: double

function `Distance`

params: x1, y1, x2, y2, phi (all doubles)

returns: double

function `Distance`

params: semi-major 1, semi-minor 1, true anomaly 1, semi-major 2, semi-minor 2, true anomaly 2, phi (all doubles)

returns: double

5 Assignment Notes

- You will receive no points if your solution does not compile on Mimir.
- Points will be deducted if your solution has any compiler warnings.
- The `<cmath>` library contains a number of useful methods including `pow`, `sqrt`, and a variety of trig functions.
- You will need to convert from degrees to radians. You will need to use the `M_PI` constant in `<cmath>`. Remember that $2\pi = 360°$.
- The `fixed` and `scientific` functions in `<iostream>` and the `setprecision` function in `<iomanip>` may be useful for helping to format your output.
- You *do not* need to check for bad input values for this project. In future projects we will explicitly indicate the errors we are looking for.
- When Mars is at perihelion (the closest point when $\theta = 0°$ and Earth is at aphelion (the farthest point when $\theta = 180°$) it is possible for the two planets to be only 5.46×10^7 km appart. This requires them to be out of phase by $\phi = 180°$.