# ERIRA Cygnus Project Math (Full 3D Version)

Nick Konz

August 15, 2019

## 1 What You Know

Given:

1. $r_c = 8.1$ kpc (distance from Earth to Galactic Center).
2. $v = 220$ km/s (tangential/rotational velocity of Earth around the galaxy.
3. Galactic longitude $\ell$ of our observations (taken from MW data)
4. Observed angular separation $\delta$ between blue and green arms
5. Observed wavelengths of green and blue arms: $\lambda_b$, $\lambda_g$, as well as the rest-frame hydrogen wavelength, $\lambda_0 = 1420.41$ MHz.

Assumptions/Approximations:

1. Tangential velocity of the blue and green and green arm regions of interest are also $v$.

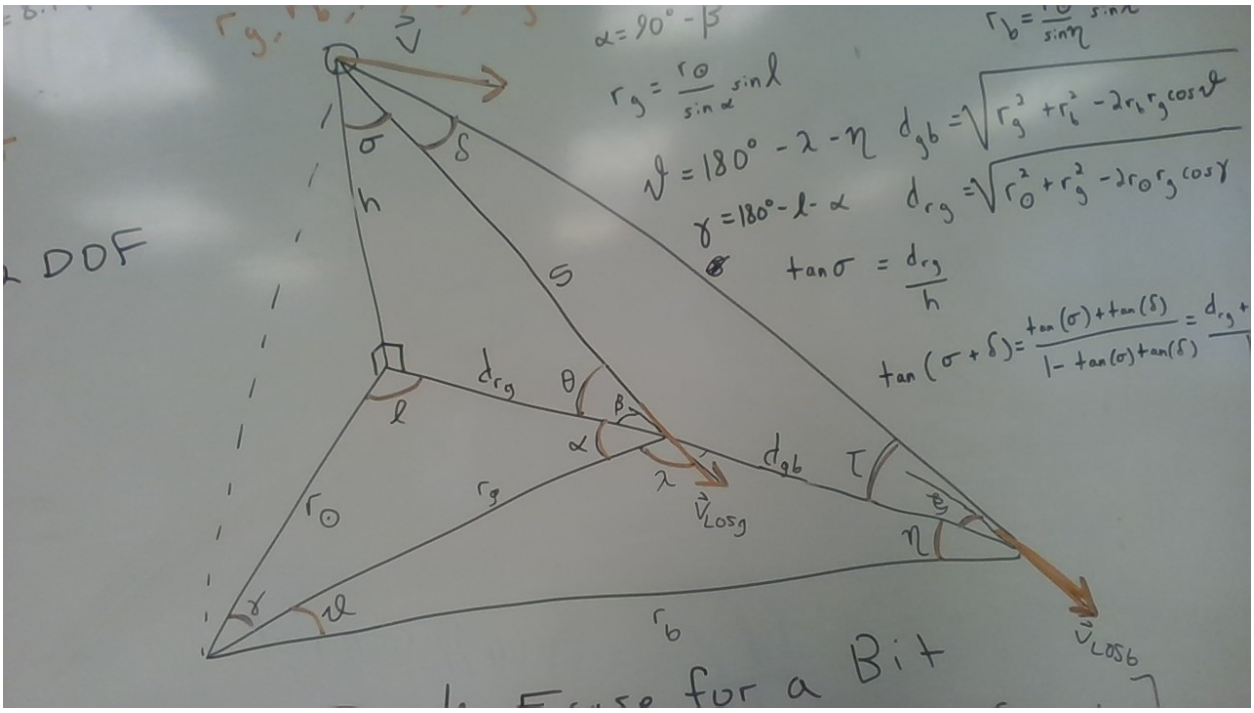


Figure 1: Geometry of the Problem

## 2 Setting Up the Problem

Observe the following diagram that illustrates the geometry of the full problem, fig. 1. Note the labeled angles and distances, and that the goal is to find the height of us above the galactic plane (assuming a flat Milky Way), $h$. From here, we can first use Doppler shift to find

$$c\frac{\lambda_g - \lambda_0}{\lambda_0} = v_{LOS,g} = v\cos\beta\cos\theta \qquad c\frac{\lambda_b - \lambda_0}{\lambda_0} = v_{LOS,b} = v\cos\xi\cos\tau. \tag{1}$$

Also note that

$$\tan\theta = \frac{h}{d_{rg}} \qquad \tan\tau = \frac{h}{d_{rg} + d_{gb}}. \tag{2}$$

Considering the horizontal plane, use basic trig to find

$$\alpha = \frac{\pi}{2} - \beta \qquad \eta = \frac{\pi}{2} - \xi \qquad \lambda = \pi - \alpha \qquad \vartheta = \pi - \lambda - \eta \qquad \gamma = \pi - \ell - \alpha, \tag{3}$$

and then the Law of Sines to obtain

$$r_g = \frac{r_c}{\sin\alpha}\sin\ell \qquad r_b = \frac{r_c}{\sin\eta}\sin\lambda. \tag{4}$$

Using the Law of Cosines, you then get that

$$d_{gb} = \sqrt{r_g^2 + r_b^2 - 2r_b r_g\cos\vartheta} \qquad d_{rg} = \sqrt{r_c^2 + r_g^2 - 2r_c r_g\cos\gamma}. \tag{5}$$

Next, look at the vertical plane. Observe that

$$\tan\sigma = \frac{d_{rg}}{h} \qquad \tan(\sigma+\delta) = \frac{d_{rg}+d_{gb}}{h}. \tag{6}$$

Use the obscure trig identity of $\tan(\sigma+\delta) = \dfrac{\tan(\sigma)+\tan(\delta)}{1-\tan(\sigma)\tan(\delta)}$ with the equation on the right of (6), you can then plug in the left of (6) into that, separate things into powers of $h$, and you'll get a quadratic equation for $h$. Solve this with the quadratic formula to obtain

$$h = \frac{d_{gb} \pm \sqrt{d_{gb}^2 - 4\tan^2(\delta)(d_{rg}+d_{gb})d_{rg}}}{2\tan(\delta)}. \tag{7}$$

# 3 The Hard Part

In the way that we normally approach this, we assume that the line of sight (LOS) velocity vectors that we're observing of the arms lie within the plane of the Milky Way (a paradoxical assumption considering that the whole point of this is to find our height above the galaxy). This version is harder because we DON'T make this assumption (if we did, $\theta$ and $\tau$ in equation (1) would just be zero). As such, finding $\theta$ and $\tau$ are required, and yet from equation (2), these angles rely on $h$, the very thing that we're trying to find. This is where it gets harder.

What you need to do is get equations (5) only in terms of $h, d_{rg}$ and $d_{gb}$, by starting from equation (1) and working downwards. You'll need to use the fact that

$$\beta = \arccos\left[\frac{v_{LOS,g}}{v\cos\left(\arctan\left[\dfrac{h}{d_{rg}}\right]\right)}\right] \qquad \xi = \arccos\left[\frac{v_{LOS,b}}{v\cos\left(\arctan\left[\dfrac{h}{d_{rg}+d_{gb}}\right]\right)}\right]. \tag{8}$$

From equation (7), you can then write $h$ in terms of $h, d_{rg}$ and $d_{gb}$, such that (5) are solely in terms of $d_{rg}$ and $d_{gb}$. Finally, solve equations (5) numerically; you have these two equations and two unknowns of $d_{rg}$ and $d_{gb}$, so they are solvable (I just plotted them both in $d_{rg}$ vs. $d_{gb}$ in Python). The way that I actually implemented this is by plotting

$$\sqrt{r_g^2 + r_b^2 - 2r_b r_g\cos\vartheta} - d_{gb} \tag{9}$$

and

$$\sqrt{r_c^2 + r_g^2 - 2r_c r_g\cos\gamma} - d_{rg}, \tag{10}$$

and seeing where they intersected. Plug in these values of $d_{rg}$ and $d_{gb}$ into (7), and solve for $h$. My code (sorry for any small mistakes) is as follows:

# 4 Python 3 Implementation

```
#----- Author: Nick Konz -----#
import argparse
from argparse import RawTextHelpFormatter
import numpy as np
import astropy as ap
import scipy as sp
import matplotlib.pyplot as plt

rC = 8.1 #kpc
v = 220 #km/s
l = np.radians(83)
delta = np.radians(5)
c = 3e5

fN = 1420.41e6 #normal
fR = 1420.5e6 #wavelengths for R, G, B
fG = 1420.71e6
fB = 1420.85e6

def w(f):
   return (3e8)/f

wN = w(fN)
wR = w(fR)
wG = w(fG)
wB = w(fB)

pm = 1;

def wDiff(w):
   return w - wN

vLOSg = c * wDiff(wG) / wN
vLOSb = c*wDiff(wB) / wN


def h(dRG, dGB):
   return (dRG + dGB - dRG + pm*np.sqrt(dGB**2 - 4*(dRG + dGB)*dRG*(np.
tan(delta)**2))) / 2*np.tan(delta)

def beta(dRG, dGB):
   return np.arccos(vLOSg/(v*np.cos(np.arctan(h(dRG, dGB)/dRG))))

def alpha(dRG, dGB):
```

```
    return np.pi/2 - beta(dRG, dGB)

def gamma(dRG, dGB):
    return np.pi - 1 - alpha(dRG, dGB)

def xi(dRG, dGB):
    return np.arccos(vLOSb/(v*np.cos(np.arctan(h(dRG, dGB)/(dRG+dGB)))))

def eta(dRG, dGB):
    return np.pi/2 - xi(dRG, dGB)

def lambd(dRG, dGB):
    return np.pi - alpha(dRG, dGB)

def rB(dRG, dGB):
    return rC*np.sin(lambd(dRG, dGB))/np.sin(eta(dRG, dGB))

def rG(dRG, dGB):
    return rC*np.sin(l)/np.sin(alpha(dRG, dGB))

def vtheta(dRG, dGB):
    return np.pi - lambd(dRG, dGB) - eta(dRG, dGB)

def f(dRG, dGB): # = 0
    return np.sqrt(rC**2 + rG(dRG, dGB)**2 - 2*rC*rG(dRG, dGB)*np.cos(
gamma(dRG, dGB))) - dRG

def g(dRG, dGB): # = 0
    return np.sqrt(rG(dRG, dGB)**2 + rB(dRG, dGB)**2 - 2*rB(dRG, dGB)*rG(
dRG, dGB)*np.cos(vtheta(dRG, dGB))) - dGB

def main():

    x = np.linspace(-50,50,1000)
    y = np.linspace(-50,50,1000)
    X, Y = np.meshgrid(x,y)

    F = f(X,Y)
    G = g(X,Y)
    Z = h(X,Y)

    plt.figure(1)
    plt.imshow(Z, extent=[-50, 50, -50, 50], origin='lower',
    cmap='viridis', alpha=0.8)
    plt.colorbar(label="height␣above␣Milky␣Way␣[kPC]")
```

```
    plt.axis(aspect='image');
    plt.contour(X,Y,F,[0], label="Condition␣1")
    plt.contour(X,Y,G,[0], cmap=plt.cm.hot, label="Condition␣2")
    plt.title("Fit␣for␣Earth's␣Height␣Above␣a␣Flat␣Milky␣Way")
    plt.ylabel('$d_{rg}$␣(distance␣from␣Earth␣to␣green␣arm␣[kPC])')
    plt.xlabel('$d_{gb}$␣(distance␣from␣green␣arm␣to␣blue␣arm␣[kPC])')
    plt.legend()
    plt.yscale('log')
    plt.show()

main()
```