Modeling Felix Baumgartner's Epic Free Fall with Differential Equations, Physics, and Computer Science

by Zachary Meeks

Felix Baumgartner about to step/jump out of his capsule—about 39,000 meters above earth!

► First, "The Givens" :

$F = ma = mg - kv^2$

where $k = D/v^2$

where $D = \frac{\alpha * \rho * v^2 * A}{2}$

α is the experimental drag coefficient

ρ is air density

A is area exposed to air-drag

where **ρ** = $\frac{P}{0.2869 * (T + 273.1)}$

P is air pressure (see C++ code and/or NASA atmospheric model)

T is temperature (see C++ code and/or NASA atmospheric model)

► Now, to solve:

What is Felix's velocity **v** = ds/dt at height h (where h = $s_0 - \Delta s$) ?

$$ds/dt = v = \int a\,dt = \int \left(\frac{dv}{dt}\right) dt$$

$$\frac{ma = mg - kv^2}{m} \rightarrow a = g - \frac{kv^2}{m}$$

$$\text{let } \lambda = \frac{k}{m}$$

$$a = dv/dt = g - \lambda v^2$$

$$\frac{dv}{g - \lambda v^2} = dt \rightarrow \int \left(\frac{dv}{g - \lambda v^2}\right) = \int dt = \int \left(\frac{dv}{g\left(1 - \left(\sqrt{\lambda/g} * v\right)^2\right)}\right)$$

$$\rightarrow \quad t = \sqrt{\frac{1}{\lambda g}}\,\text{artanh}\left(\sqrt{\frac{\lambda}{g}}v\right) \quad | \textit{ evaluated from } v_0 \textit{ to } v(t)$$

let $t_* = t + t_0 = t_{v(t)}$

$$t_* \sqrt{\lambda g} = \text{artanh}\left(\sqrt{\frac{\lambda}{g}}v\right)$$

$$\rightarrow \quad \tanh\left(t_* * \sqrt{\lambda g}\right) = \sqrt{\frac{\lambda}{g}}v$$

$$\sqrt{\frac{g}{v}}\tanh\left(t_* \sqrt{\lambda g}\right) = v$$

$$v = \frac{ds}{dt} = \sqrt{\frac{g}{\lambda}}\frac{\sinh\left(t_* \sqrt{\lambda g}\right)}{\cosh\left(t_* \sqrt{\lambda g}\right)}$$

$$\rightarrow \quad \int ds = \int \left(\sqrt{\frac{g}{\lambda}}\frac{\sinh\left(t_* \sqrt{\lambda g}\right)}{\cosh\left(t_* \sqrt{\lambda g}\right)}\right) dt$$

$$\rightarrow \quad s = \sqrt{\frac{1}{\lambda^2}}\ln\left(\cosh\left(t_* \sqrt{\lambda g}\right)\right) \quad | \textit{ evaluated from } t_{*0} \textit{ to } t(s)$$

let $s_* = s + s_{*0} = s_{t(s)}$

→ $s_* = \sqrt{\frac{1}{\lambda^2}} \ln(\cosh(t_* \sqrt{\lambda g}))$

$\sqrt{\lambda^2}\ s_* = \ln(\cosh(t_* \sqrt{\lambda g}))$

→ $e^{\lambda s_*} = e^{\ln(\cosh(t_* \sqrt{\lambda g}))} = \cosh(t_* \sqrt{\lambda g})$

$\operatorname{arcosh}(e^{\lambda s_*}) = t_* \sqrt{\lambda g}$

$t_* = \frac{\operatorname{arcosh}(e^{\lambda s_*})}{\sqrt{\lambda g}} = t + t_0$

→ $t = \frac{\operatorname{arcosh}(e^{\lambda s_*})}{\sqrt{\lambda g}} - t_0$

We should be able to thus determine the intervals of t corresponding to $h = s_0 - \Delta s$ from the above equations. We can then find the values of v in intervals corresponding to the values of s and the found values of t.

In the above equations, **s** is essentially an independent variable in that we will be solving for v based on arbitrary values of s.

The complexity of the above equations is called for because of the drag equation's necessity of height in the calculation of temperature. If height wasn't a factor, the problem of finding Felix's velocity would just be a first-order differential equation.

Next, I provide my C++ code, followed by the data generated/output by the code, and then a graph that corresponds to that data.

Then I provide Two of Nasa's atmospheric models, followed by two graphs that my model generated that correspond to the earth's atmospheric model.

Then I provide data for a hypothetical jump from 49 kilometers.

I end with notes and observations on my model.

```
#include <QCoreApplication>
#include <iostream>
#include <cstdlib>
#include <cmath>

using namespace std;

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);

    double alpha, rho, A, gamma, mass;
    double T1, T2, T3, P1, P2, P3;
    double t0, t1, HH, g, v0, v1, C_vt, C_ts;
    double m3, m00, m2, m1, m007, sigma_t, delta_t, v_mesh;
    int h;

    cout<<"what initial height will Felix be falling from? (in kilometers)"<<endl;
    cin>>h;
    h=h*1000;

    //alpha is the experimental drag coefficient
    alpha= .93;     //set to 0.93 to get the right numbers---experimental drag
                    //coeff after all

    //A is the area (in meters sqrd.) that is exposed to drag.  Note: here, set up
    //variably
    A= 0.33;

    //this was the mass of Felix with his equipment on
    mass = 117.5;

    //let g = 9.8 m/s^2 (consider gravity constant)
    g=9.8;

    t0= 0;   //time

    delta_t=20;
    sigma_t=0;

    HH=delta_t;  //counting height from nought-point

    v0=0;

    while(h>25000)
    {
        T3= -131.21 + .00299*h;

        m3= (T3 + 273.1)/216.6;

        P3= 2.488 * pow(m3, -11.388);

        rho= P3/(.2869*(T3+273.1));
```

```
    gamma= (alpha*rho*A)/(2*mass);

    m00= exp(gamma*HH);

    t1=  acosh(m00)/sqrt(gamma*g);

    h=h-(int)delta_t;

    v1= sqrt(g/gamma)*tanh(t1*sqrt(gamma*g));

    v_mesh=(v0+v1)/2.0;

    sigma_t = sigma_t + delta_t/v_mesh;

    if(h%1000==0)
        cout<<"TIME = "<<sigma_t<<"sec"<<endl;

    v0=v1;

    HH= HH + delta_t;

    if(A<.57 && h%1000==0)
        A=A+.01155;

    if(A<.57 && h <30000 && h%1000==0)
        A=A+.004505;

    if(h%1000==0)
        cout<<"      Felix is going "<<v1<<" m/s at height "<<h/1000<< "km
        above earth"<<endl;
}

t0=t1;

//C_vt = sqrt(1.0/(gamma*g))*atanh(sqrt(gamma/g)*v1);  //not important,
//evaluates to same as t0

m007= pow(gamma, -2.0);

C_ts= sqrt(m007)*log(cosh(t1*sqrt(gamma*g)));

HH=delta_t;

while(h>11000)
{
    T2= -56.46;
    m2= 1.73-0.000157*h;
    P2= 22.65*exp(m2);
    rho= P2/(.2869*(T2+273.1));
    gamma= (alpha*rho*A)/(2*mass);

    m00= exp(gamma*(HH+C_ts));
```

```
    t1= acosh(m00)/sqrt(gamma*g);

    h=h-(int)delta_t;

    v1= sqrt(g/gamma)*tanh((t1)*sqrt(gamma*g));

    v_mesh=(v0+v1)/2.0;

    sigma_t = sigma_t + delta_t/v_mesh;

    if(h%1000==0)
        cout<<"TIME = "<<sigma_t<<"sec"<<endl;

    v0=v1;

    HH=HH + delta_t;

    if(A<.715 && h%1000==0)
        A=A+.01475;

    if(A>.714 )
        A=.71025;

    if(h%1000==0)
        cout<<"      Felix is going "<<v1<<" m/s at height "<<h/1000<< "km
        above earth"<<endl;
}

t0=t1;

//C_vt = sqrt(1.0/(gamma*g))*atanh(sqrt(gamma/g)*v1);
//^^^even if this was important, it appears atanh to tanh loses too many sig
//figs, such that it would be pretty useless

m007= pow(gamma, -2.0);

C_ts= sqrt(m007)*log(cosh(t1*sqrt(gamma*g)));

HH=delta_t;

A= 0.715;

while(h>0)
{
    T1= 15.04 - .00649*h;
    m1= (T1+273.1)/288.08;
    P1= 101.29*pow(m1, 5.256);
    rho= P1/(.2869*(T1+273.1));
    gamma= (alpha*rho*A)/(2*mass);

    m00= exp(gamma*(HH+C_ts));

    t1= acosh(m00)/sqrt(gamma*g);
```

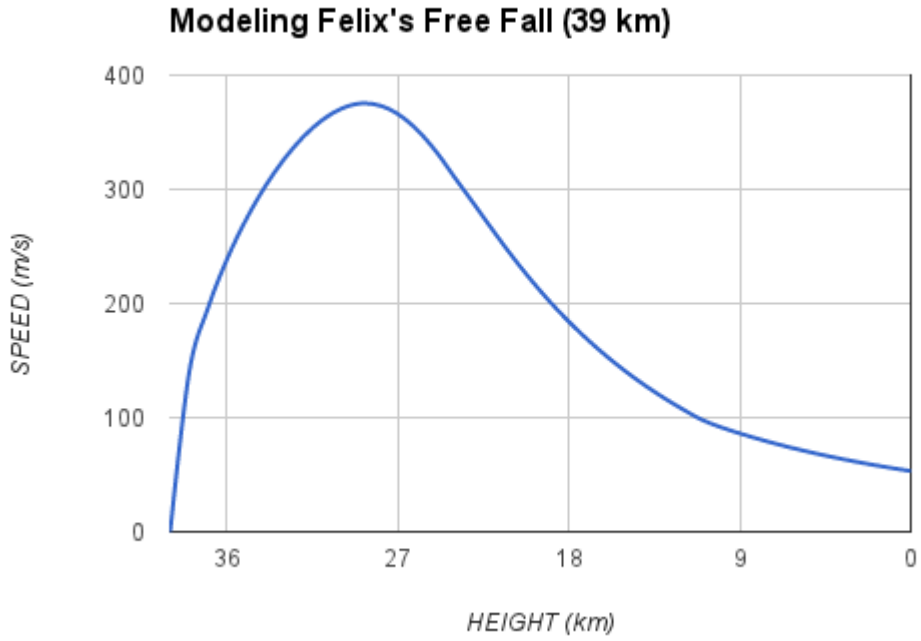
```
        v1= sqrt(g/gamma)*tanh((t1)*sqrt(gamma*g));

        v_mesh=(v0+v1)/2.0;

        sigma_t = sigma_t + delta_t/v_mesh;

        h=h-(int)delta_t;

        if(h%1000==0)
            cout<<"TIME = "<<sigma_t<<"sec"<<endl;

        v0=v1;

        HH=HH + delta_t;

        if(h%1000==0)
            cout<<"      Felix is going "<<v1<<" m/s at height "<<h/1000<< "km
            above earth"<<endl;

    }

    return a.exec();
}
```

Felix Free Fall Data

(C++ code output for 39km free fall)

TIME = 0sec
    Felix is going 0 m/s at height 39km above earth
TIME = 14.301sec
    Felix is going 139.53 m/s at height 38km above earth
TIME = 20.2511sec
    Felix is going 196.404 m/s at height 37km above earth
TIME = 24.8425sec
    Felix is going 239.015 m/s at height 36km above earth
TIME = 28.7425sec
    Felix is going 273.636 m/s at height 35km above earth
TIME = 32.2132sec
    Felix is going 302.479 m/s at height 34km above earth
TIME = 35.3927sec
    Felix is going 326.447 m/s at height 33km above earth
TIME = 38.3674sec
    Felix is going 345.841 m/s at height 32km above earth
TIME = 41.1987sec
    Felix is going 360.618 m/s at height 31km above earth
TIME = 43.9349sec
    Felix is going 370.526 m/s at height 30km above earth
TIME = 46.6183sec
    Felix is going 375.208 m/s at height 29km above earth
TIME = 49.2891sec
    Felix is going 374.31 m/s at height 28km above earth
TIME = 51.9885sec
    Felix is going 367.616 m/s at height 27km above earth
TIME = 54.7611sec
    Felix is going 355.203 m/s at height 26km above earth
TIME = 57.6562sec
    Felix is going 337.599 m/s at height 25km above earth
TIME = 60.9358sec
    Felix is going 295.628 m/s at height 24km above earth
TIME = 64.4599sec
    Felix is going 274.185 m/s at height 23km above earth
TIME = 68.273sec
    Felix is going 252.74 m/s at height 22km above earth
TIME = 72.4177sec
    Felix is going 232.201 m/s at height 21km above earth
TIME = 76.9326sec
    Felix is going 213.044 m/s at height 20km above earth
TIME = 81.8544sec
    Felix is going 195.401 m/s at height 19km above earth

TIME = 87.2202sec
Felix is going 179.223 m/s at height 18km above earth
TIME = 93.0698sec
Felix is going 164.403 m/s at height 17km above earth
TIME = 99.4459sec
Felix is going 150.826 m/s at height 16km above earth
TIME = 106.395sec
Felix is going 138.387 m/s at height 15km above earth
TIME = 113.968sec
Felix is going 126.987 m/s at height 14km above earth
TIME = 122.22sec
Felix is going 116.54 m/s at height 13km above earth
TIME = 131.211sec
Felix is going 106.964 m/s at height 12km above earth
TIME = 141.006sec
Felix is going 98.1843 m/s at height 11km above earth
TIME = 151.591sec
Felix is going 91.5768 m/s at height 10km above earth
TIME = 162.852sec
Felix is going 86.1561 m/s at height 9km above earth
TIME = 174.812sec
Felix is going 81.1942 m/s at height 8km above earth
TIME = 187.492sec
Felix is going 76.6413 m/s at height 7km above earth
TIME = 200.915sec
Felix is going 72.454 m/s at height 6km above earth
TIME = 215.103sec
Felix is going 68.5945 m/s at height 5km above earth
TIME = 230.079sec
Felix is going 65.0298 m/s at height 4km above earth
TIME = 245.866sec
Felix is going 61.731 m/s at height 3km above earth
TIME = 262.485sec
Felix is going 58.6723 m/s at height 2km above earth
TIME = 279.961sec
Felix is going 55.8314 m/s at height 1km above earth
TIME = 298.315sec
Felix is going 53.1882 m/s at height 0km above earth


Modeling Felix's Free Fall (39 km)
400
300
200
100
0
SPEED (m/s)
36
27
18
9
0
HEIGHT (km)

Earth Atmosphere Model

Metric Units

Glenn Research Center

For $h > 25000$ (Upper Stratosphere)

$$T = -131.21 + .00299\,h$$

$$p = 2.488 * \left[\frac{T + 273.1}{216.6}\right]^{-11.388}$$

For $11000 < h < 25000$ (Lower Stratosphere)

$$T = -56.46$$

$$p = 22.65 * e^{(1.73 - .000157\,h)}$$

For $h < 11000$ (Troposphere)

$$T = 15.04 - .00649\,h$$

$$p = 101.29 * \left[\frac{T + 273.1}{288.08}\right]^{5.256}$$

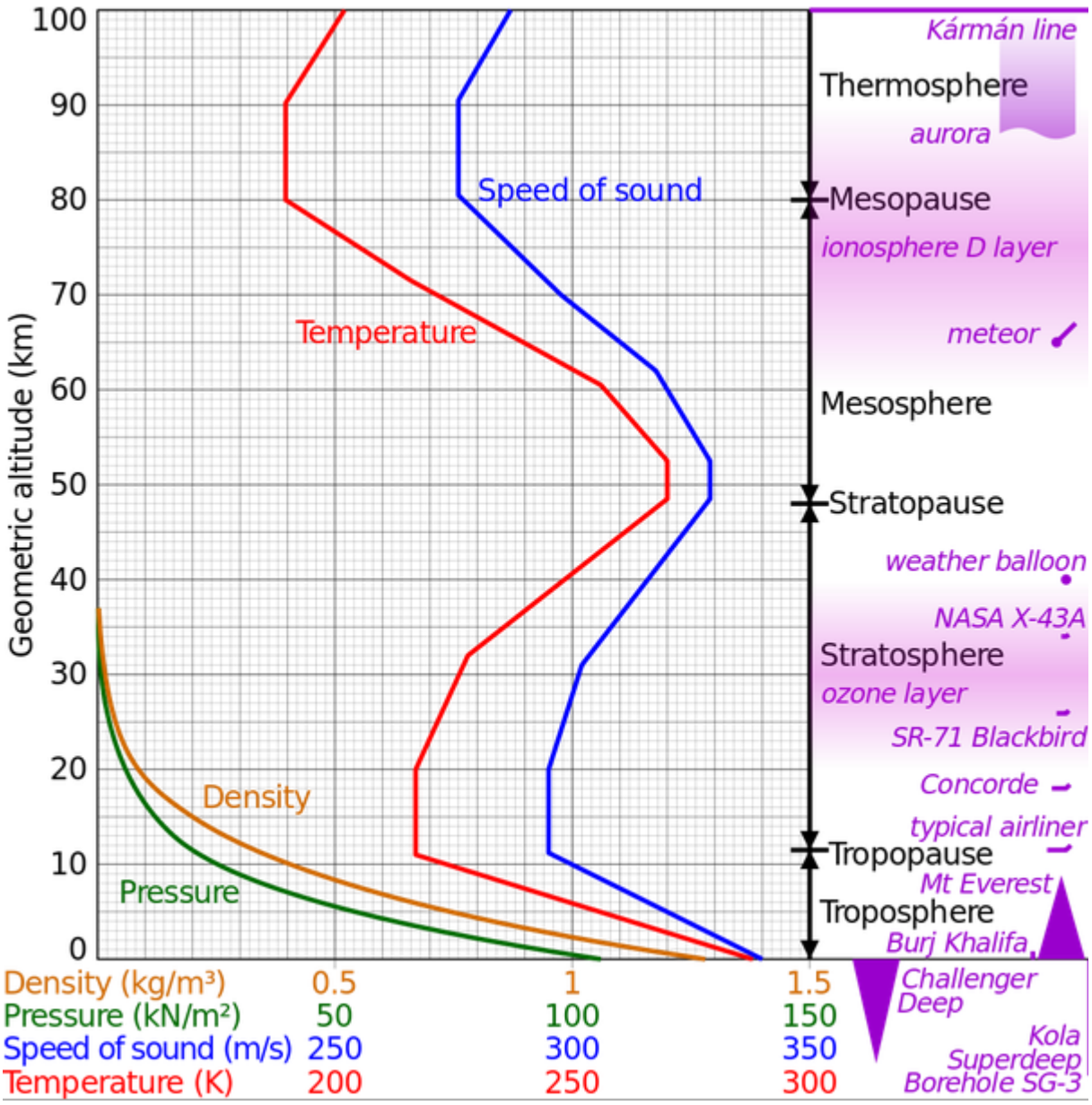
ρ = density (kg/cu m)

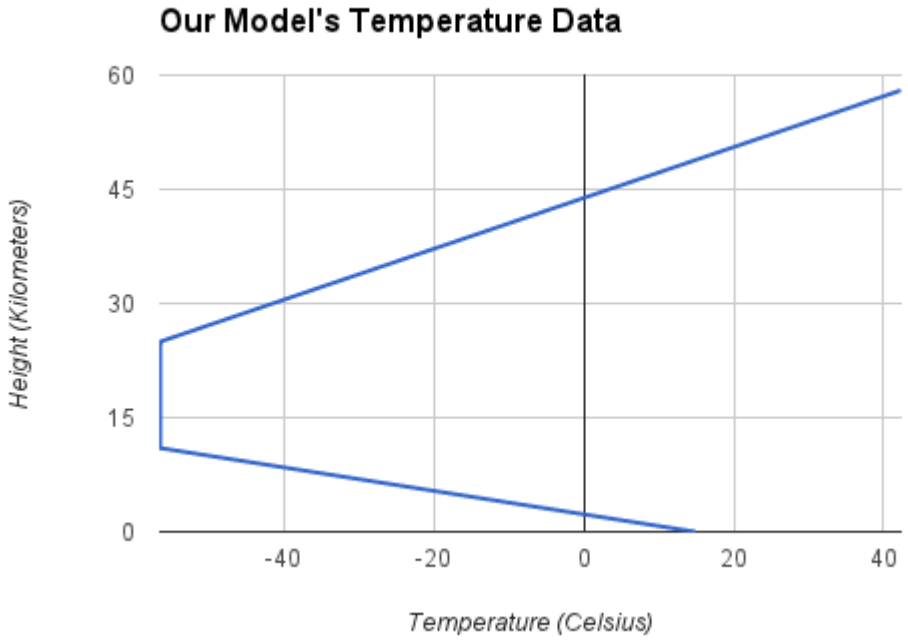
p = pressure (K-Pa)

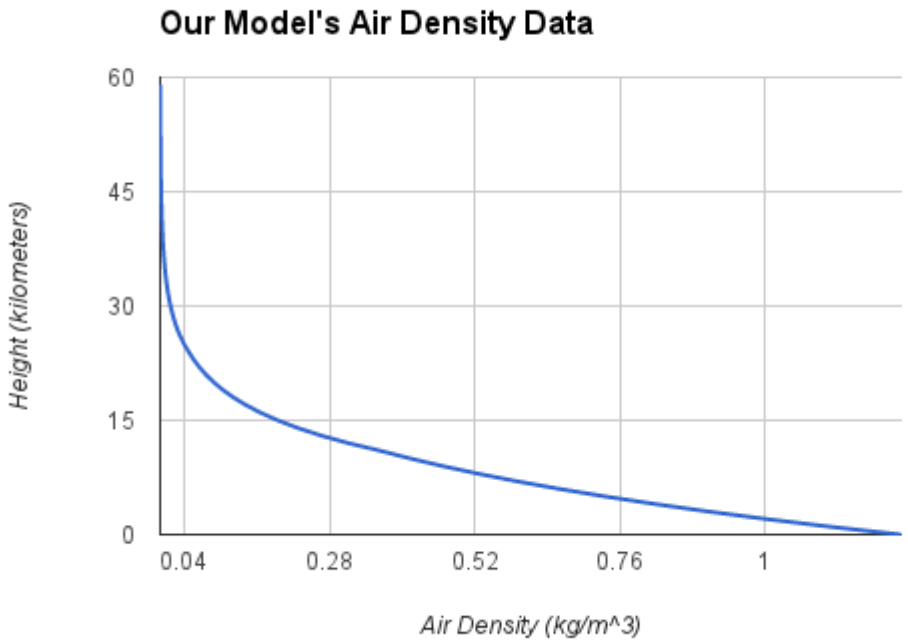
$$\rho = p / (.2869 * (T + 273.1))$$

T = temperature (^{0}C)

h = altitude (m)


Geometric altitude (km)
100
90
80
70
60
50
40
30
20
10
0
Temperature
Speed of sound
Density
Pressure
Kármán line
Thermosphere
aurora
Mesopause
ionosphere D layer
meteor
Mesosphere
Stratopause
weather balloon
NASA X-43A
Stratosphere
ozone layer
SR-71 Blackbird
Concorde
typical airliner
Tropopause
Mt Everest
Troposphere
Burj Khalifa
Challenger Deep
Kola Superdeep Borehole SG-3
Density (kg/m³) 0.5 1 1.5
Pressure (kN/m²) 50 100 150
Speed of sound (m/s) 250 300 350
Temperature (K) 200 250 300


Our Model's Temperature Data
60
45
30
15
0
Height (Kilometers)
-40
-20
0
20
40
Temperature (Celsius)


Our Model's Air Density Data
60
45
30
15
0
Height (kilometers)
0.04
0.28
0.52
0.76
1
Air Density (kg/m^3)

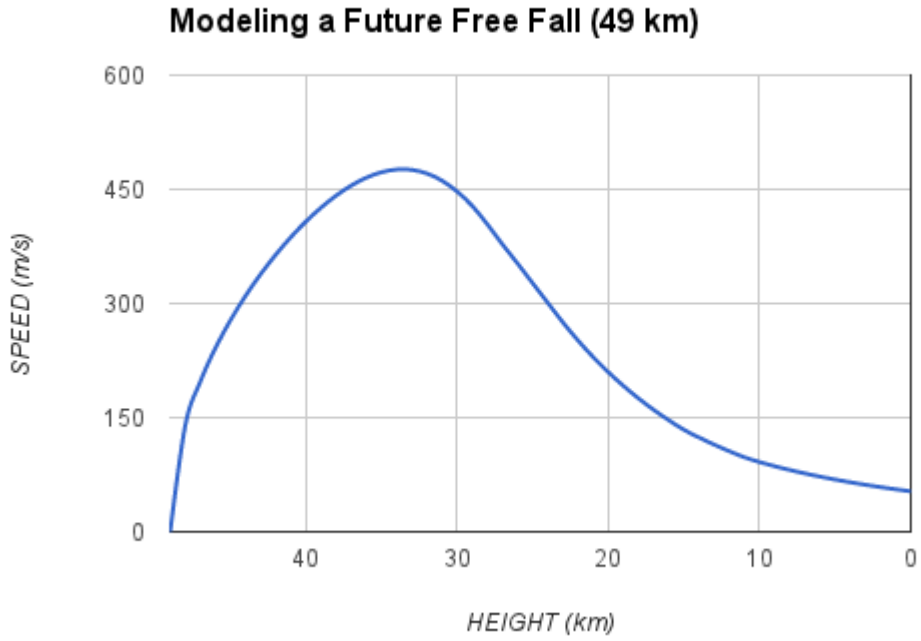
Hypothetical Free Fall Data

(C++ code output for 49km free fall)

TIME = 0sec
    Felix is going 0 m/s at height 49km above earth
TIME = 14.2896sec
    Felix is going 139.88 m/s at height 48km above earth
TIME = 20.2152sec
    Felix is going 197.591 m/s at height 47km above earth
TIME = 24.7683sec
    Felix is going 241.624 m/s at height 46km above earth
TIME = 28.6138sec
    Felix is going 278.436 m/s at height 45km above earth
TIME = 32.0097sec
    Felix is going 310.475 m/s at height 44km above earth
TIME = 35.0893sec
    Felix is going 338.942 m/s at height 43km above earth
TIME = 37.9324sec
    Felix is going 364.491 m/s at height 42km above earth
TIME = 40.5922sec
    Felix is going 387.474 m/s at height 41km above earth
TIME = 43.1064sec
    Felix is going 408.059 m/s at height 40km above earth
TIME = 45.5037sec
    Felix is going 426.276 m/s at height 39km above earth
TIME = 47.8074sec
    Felix is going 442.049 m/s at height 38km above earth
TIME = 50.0369sec
    Felix is going 455.204 m/s at height 37km above earth
TIME = 52.2099sec
    Felix is going 465.487 m/s at height 36km above earth
TIME = 54.3429sec
    Felix is going 472.571 m/s at height 35km above earth
TIME = 56.4524sec
    Felix is going 476.073 m/s at height 34km above earth
TIME = 58.5557sec
    Felix is going 475.588 m/s at height 33km above earth
TIME = 60.6713sec
    Felix is going 470.73 m/s at height 32km above earth
TIME = 62.8204sec
    Felix is going 461.2 m/s at height 31km above earth
TIME = 65.0267sec
    Felix is going 446.874 m/s at height 30km above earth
TIME = 67.3183sec
    Felix is going 427.905 m/s at height 29km above earth

TIME = 69.7337sec
    Felix is going 403.619 m/s at height 28km above earth
TIME = 72.3041sec
    Felix is going 377.278 m/s at height 27km above earth
TIME = 75.0463sec
    Felix is going 352.13 m/s at height 26km above earth
TIME = 77.9958sec
    Felix is going 326.229 m/s at height 25km above earth
TIME = 81.202sec
    Felix is going 300.432 m/s at height 24km above earth
TIME = 84.7043sec
    Felix is going 274.724 m/s at height 23km above earth
TIME = 88.5362sec
    Felix is going 250.995 m/s at height 22km above earth
TIME = 92.7302sec
    Felix is going 229.297 m/s at height 21km above earth
TIME = 97.3201sec
    Felix is going 209.516 m/s at height 20km above earth
TIME = 102.342sec
    Felix is going 191.491 m/s at height 19km above earth
TIME = 107.835sec
    Felix is going 175.061 m/s at height 18km above earth
TIME = 113.843sec
    Felix is going 160.08 m/s at height 17km above earth
TIME = 120.411sec
    Felix is going 146.416 m/s at height 16km above earth
TIME = 127.59sec
    Felix is going 133.948 m/s at height 15km above earth
TIME = 135.359sec
    Felix is going 123.796 m/s at height 14km above earth
TIME = 143.762sec
    Felix is going 114.45 m/s at height 13km above earth
TIME = 152.852sec
    Felix is going 105.809 m/s at height 12km above earth
TIME = 162.684sec
    Felix is going 97.8207 m/s at height 11km above earth
TIME = 173.269sec
    Felix is going 91.5768 m/s at height 10km above earth
TIME = 184.53sec
    Felix is going 86.1561 m/s at height 9km above earth
TIME = 196.49sec
    Felix is going 81.1942 m/s at height 8km above earth
TIME = 209.17sec
    Felix is going 76.6413 m/s at height 7km above earth
TIME = 222.593sec
    Felix is going 72.454 m/s at height 6km above earth
TIME = 236.781sec
    Felix is going 68.5945 m/s at height 5km above earth
TIME = 251.757sec

Felix is going 65.0298 m/s at height 4km above earth
TIME = 267.544sec
Felix is going 61.731 m/s at height 3km above earth
TIME = 284.163sec
Felix is going 58.6723 m/s at height 2km above earth
TIME = 301.639sec
Felix is going 55.8314 m/s at height 1km above earth
TIME = 319.993sec
Felix is going 53.1882 m/s at height 0km above earth


Modeling a Future Free Fall (49 km)
600
450
300
150
0
SPEED (m/s)
40
30
20
10
0
HEIGHT (km)

Notes about my model:

I chose to [midpoint] Riemann sum the calculation of time. The C++ code correctly calculates time for about the first 13,000 meters, then it starts diverging. However, the velocity calculations continued to be what I would expect, so I switched all of the time calculations, as output, to be based off of the velocity calculations, which oddly enough are based on the erroneous time calculations. I haven't yet figured out why the velocity calculations don't similarly diverge. Some things to consider in clarifying why the velocity values remain to be what we would expect include:

-The choice of experimental drag coefficient may be a somewhat self-fulfilling prophecy

-As time t increases, the values of variables play a larger role in calculating other variables' values, all of which will lead to a compounding effect of truncation error

-tanh and arctanh don't switch well between eachother without creating a large-number format (because of how quickly tanh approaches 1), which I haven't written or enabled

My data versus the actual Red Bull Stratos data:

Actual max speed:	377 m/s
My model:	375 m/s
Actual h of max speed:	30 or 31 km
My model:	29 km
Actual flight time:	4:20
my model:	4:06 (assuming exactly 36km of free falling)

Other Considerations:

To provide for a smoother graph, and assuming that Felix would want to be aerodynamic until he hit max speed, at which point he would then want to start slowing down, I programmed the model to have him smoothly transition from around A= .33 meters-squared to A=.715 meters-squared over the course of his flight. This variable has a significant role in the program, and if I started it at 0.715 instead, my model would clock a total flight time of about 4:12. However, max speed would then reduce to about 345 m/s. My model is a simple model. There are myriad other unaccounted for variables, many of which are quite significant factors, such as temperature, air pressure, and wind. It should be noted that the Red Bull Stratos' model predicted a flight time close to five minutes. If that is true, then that would suggest that the experimental drag coefficient is perhaps larger than I assume it is, and the weather encountered offered less air density than was expected (and less than the averaged NASA atmospheric model that I used in my model). Or perhaps my model works better than theirs'. #Joke

I have provided data of a hypothetical free fall from 49 kilometers because I think it illustrates that my model can model data for experiments outside of the one that I

programmed it for. The stratopause layer is around 50 kilometers, at which point the temperature is due to change its algorithm, so my model is limited up to the stratopause.