Things I have done + notes:

* Read spectrum files
  + simple
* analyse spectrum files
  + FUCk, scipy.constants.e IS NOT THE SAME AS scipy.e
  + FUCK, those are some weird models
  + You would not believe your codes
    - If then million data points
    - Gave model that was total whack
    - Cause they drop down all the way
    - And leave questions everywhere
    - And all I can do is I would just stand and.. Stare
    - I try to make myself believe that my all my models do well
    - It’s hard to say that I’d rather work a lot when I’m asleep
    - Cause everything is never as it seems
  + OMG, JUST WORK
    - Tried manually changing the particle velocity to like 30\* normal
      * More like 20 000 times normal
    - WHY DIDN’T I JUST KEEP THE OLD THING THAT I THOUGHT MIGHT WORK BUT DOESN’T
  + Hey it worked
  + Nice
* Make density profile
  + Struggle hella hard
  + Find out that you are meant to make it more simple than that
  + Something something “fuck it, I’mma just make some assumptions”
  + Hey, it suddenly became *really* easy!
  + Who’d’ave thunk it
  + Ok, not quite sure I did it right, but it works so phucket bucket
* Hey, I got into lower orbit
  + Just straight up drop down and blasted off again (Team Rocket blasts away again)
  + Now I just have to calculate rotational velocity (haven’t we done something similar earlier?) (Also the fuck do phi and theta refer to) (asked on piazza, now we hope)
* Take picturrrreeeesss
* Had to adjust air resistance for wind
  + Not necessary, but felt like doing it proper

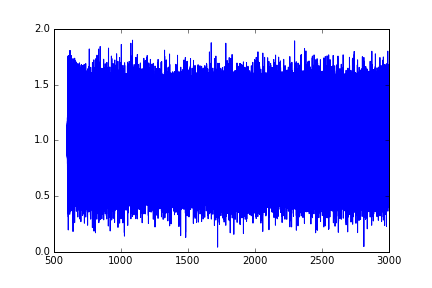
Notes on graph reading:

* CO – Yes
* CH4 – Yes
* N2O – Undecided (remember this is supposed to have life, but cold life maybe?)
* H2O – Yes
* CO, CH4
  + Easy
* Line 940
  + Model says no  
    eyes says yes  
    decide on no
  + Because other pieces of H2O (line 940 is H2O), I decide H2O is a thuing
* Difficult ones to decide:
  + CO2
  + O2
  + Do later?
* NEW DATA CAME IN
* Yes
  + CO
  + CH4
  + H2O
  + O2
* No
  + N2O
  + CO2
* HAD TO REDO ORBIT, BECAUSE OF COURSE IT WASN’T CLOSE ENOUGH FOR THE LANDING PROCEDURES TO KICK IN

Last time on Spacey Balls P, we watched Tom as he struggled to find his way to Fræfdal, the land in which he believed God to reside. This time, after redoing his orbit because *apparently he wasn’t close enough for the automatic landing to kick in*, we watch as he attempts to scan Fræfdal for gases, lower his orbit to a suitable location and take lots of pretty pictures and videos while finding a landing spot. Can he do it, or will he fall and crash multiple times while talking to people who aren’t even there or even rapping against a fucking chair. Find out in this exciting new space-themed blog post of Spacer Stalls B!

So yeah, anyways, ignore that terrible intro please. It has… *unsavoury* influences. Anyways! I really wanted to land on Fræfdal, and to do that, I needed to do a couple of things first; first of all being scanning the atmosphere to find what kind of gases there are.

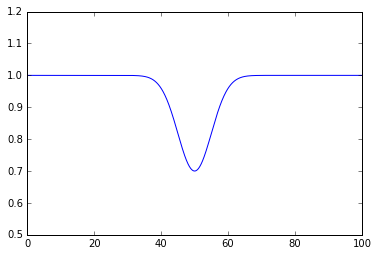
I had data (so much fucking data) showing me a range of light frequencies and how much each was registered as coming off Fræfdal, under a fuckton of disturbances and noise:



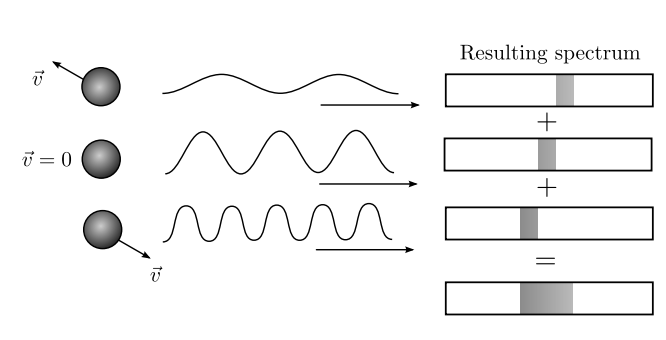
As you can see, the data is basically buried under noise

So I had to \*cracks knuckles\* drag out some real \*stretches arms\* hardcore data analysis methods \*cracks neck\* to take this beasty down!  
If by hardcore methods you mean the least squared method I mentioned and performed in an earlier blog post, that is. If you remember, lucky you! I am still going to explain it again, because now the magnitude of the noise *varies*, so there are slight differences I have to mention. Technically, these small differences means we are using a method called “least χ method” (that is a greek xhi, not an x btw).

So let’s start with how we expect the graph to actually look like! Since this is an absorption spectrum, meaning we are looking at what light frequencies the gases in the atmosphere absorb, there should be some dips in the graph showing what has been absorbed.

If we want to model this dip mathematically, we can start by having some constant that is the maximum height Fmax, and some part (Fmin – Fmax)g(x) where g(x) becomes very small when it moves away from a point, and becomes 1 when at that point. This makes it so that that point, the point of the frequency that was actually absorbed, the model becomes Fmax, + (Fmin – Fmax) = Fmin, and further away it just becomes Fmax. Here’s an example of such a graph in action, centered around a frequency of 50 (I just chose that number because why the fuck not).  


Note that the drop is not completely thin, and has some width. This is because of temperature, making particles go in random directions, causing doppler effects with both increases and decreases in wavelength. This spreads out the dip, making it look like it does.

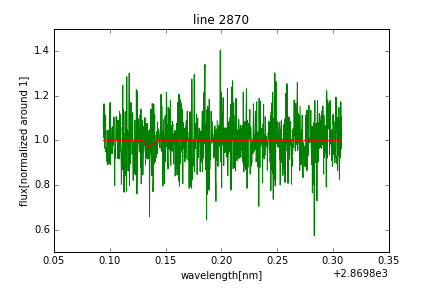
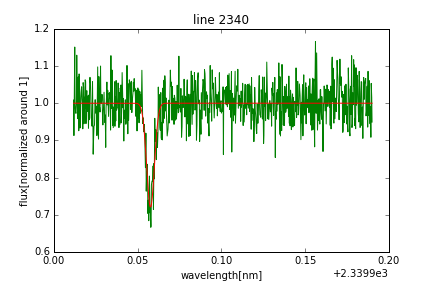
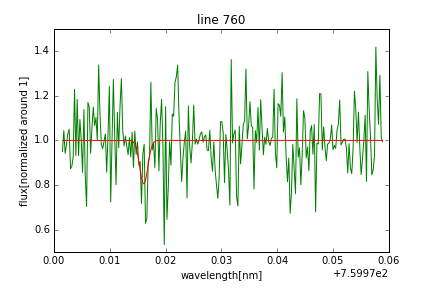
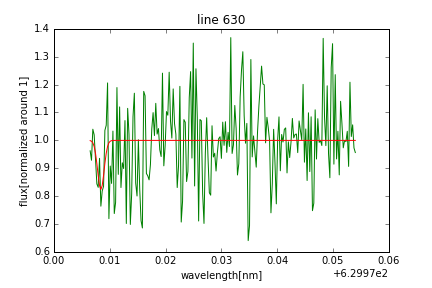


How temperature spreads spectrum with doppler effects

So now we gotta figure out what g(x) looks like. We could make something up, and feel pretty proud of ourselves, but we could also just steal someone else’s work and be lazy. I quite prefer being lazy so we steal some of the function for a gaussian bell curve, which you might have seen before, and we modify it; . This thing has the amazing ability of going towards , when , the thing that varies across the x-axis, gets close to , a constant that shows where the dip is centered. It also goes towards 0 as we get further away from the given center, as we get something to the power of a large negative number, which becomes very small. The us another constant that shows how wide the gap is. Not quite sure why we bother squaring it, but Gauss did it, so we might as well. I mean, Gauss was a very smart fellow, so I’m gonna trust him to have some clue what he was doing

So now we have a function that can model shit, now we just need to find what constants to put into that model to best fit the “graph” we’re going to check (to be honest, can we call it a graph? More like blob amirite?). We could do this by the method of least square, but the amount of noise is not constant, meaning that some of the data will be closer to the truth than other pieces of data. If we have an estimate for how much noise there is as a function of time, we can weigh the datapoints so that the points with less noise count more. I will explain how to do this when we get to it.  
Basically, the only difference between the least square method and the χ method is this variance in noise, and for some reasons mathematicians decided that that was enough to distinguish them from eachother.

So we take the model and test for some constants , and Fmin, find the difference between the model and the actual data in each point and divide it by the noise level for that point. This way the points that have lots of noise don’t count as much towards the χ value, while the ones who have little noise will be more important. Then we square those so that being very far off have more to say, because the electoral college is a good system, and this is totally related. We do this for a fuckton of different combinations of constants (like 270 000 combinations, I’m not fucking around here), and the one with the smallest total difference, χ, is the best fit:



As you can see, some are more obvious than others and I need to go through each spot where we expect a dip and see if there is one, with the model as help.

Basically, I check for the presence of 6 different gases; O2, H2O, CO2, CH4, CO, N2O. In a laboratory setting, with no doppler shifts, they show holes in the absorption spectras as such:

* O2
  + 630
  + 690
  + 760
* H2O
  + 720
  + 820
  + 940
* CO2
  + 1400
  + 1600
* CH4
  + 1660
  + 2200
* CO
  + 2340
* N2O
  + 2870

And I found signs of CO, O2, H2O and CH4, meaning I should probably be careful with matches down there.

I could try to find out how much there is of each gas… or I could just assume they all exist in equal amounts…

…

I think we all know what I’m gonna do here.

…

So now that we’ve established the atmosphere is composed of 25% carbon monoxide, 25% oxygen, 25% water and 25% methane, we can start working out some things we need to know to land there, like how dense the atmosphere is. Since this is mostly just maths, I’m gonna save it for later, but suffice to say we make some *assumptions* to simplify the problem.

We also need some miscellaneous equations, like the terminal velocity close to the surface and size of a parachute needed to achieve a certain velocity near the surface. We mostly want the parachute because crashing and dying sucks.

The terminal velocity means that the force from the air resistance, given as , and the force from gravity, given as , cancel each other out; .  
With some readjustment, this gives , Where G is the gravitational constant, M is the mass of the planet, m is the mass of the satellite, R is the distance from the planet or as we use in this instance the radius of the planet itself, is the density of the atmosphere at the surface and A is the area of the satellite/parachute.

In order to find the size of a parachute needed to reach a terminal velocity of 3 m/s, we just reorder the equation so that A is alone on the left side (allll byyy itseeeeeeelf). Apparently, I managed to fuck this up and put in the pressure P instead of the density (easy to make that mistake, ok?). Mistakenly using the pressure instead of the density gave me an area of 0.01 , which raised some alarm bells. Luckily, I applied my *expert* troubleshooting skills and found the problem almost immediately. And by that I mean that I didn’t bother fixing it until right now when I’m writing this. Nice.  
Anyways, fixing the codes gave me a parachute size of 46.7 , which makes more sense.

Now that we have those things, we can start working on our actual goal this time! Getting into lower orbit and getting badass footage while doing it! We want to go down as close as we can before air resistance starts fuckign us up and sends us spiralling in and crashing. A good spot to start could be anywhere where , so that the air resistance is negligible.   
What I did was make a simulation software, just like the one I made earlier to simulate the satellites path through the star system, except I added in the effect of air resistance.

There might be some very clever ways of decreasing altitude, but I believe I had the most clever method of all; remove all velocity with an initial boost and let yourself fall downwards, then speed up again when you’re as low as you want. I did this in my simulation, and calculated as I fell. When looked good, I gave an explosive boost that brought me into low orbit.



The most brilliant plan

It worked pretty well, I think. You can do a lot when you boost yourself with what is basically explosions and have no consideration for your own wellbeing.

[INPUT VIDEO OF FALLIGN DOWN]

In order to find out what velocity I needed to have when in low orbit, I used the handy-dandy function that I fou- \*cough\*… ehhh…. Made, I mean. Yes, I did that all by myself, I did not give in to laziness this time, no sir. Anyways, I’m go to, for reasons that may or may not be explained later, just give you the formula and not explain where it came from:

[THIS IS WHERE WE SHOW PICTURES OF SUNRISE, LANDING SPOTS, AND PANNIGN ACROSS SURFACE LOOKING FOR LANDING SPOTS]

[ALSO DON’T FORGET MATHS FOR DENSITY PROFILE]

[ALSO LONGITUDE AND ALTITUDE]

630, 690, 760, 720, 820, 940, 1400, 1600, 1660, 2200, 2340, 2870  
"O2", 3], ["H2O", 3], ["CO2", 2], ["CH4", 2], ["CO", 1], ["N2O"

Criteria:

* How well is the task solved (previous 100% )
* Explain the actual problem (previous 100% )
* Explain thought process (previous 100% )
* How the blog is built, and results (previous 100% )
* FIGURES/ILLUSTRATIONS (previous 100% )