There are 5 probems, each of which is worth 20 points. Two bonus points for following instructions, which can make up for mistakes elsewhere.

Problem 0:

Please put everything into a directory. Include a file called "myinfo.txt" that has your name and ID in it, so I can match grades to students at the end. (2 marks).

Problem 1:

You may have seen recent report that phosphine, a potential signature of life, was detected in the upper atmosphere of Venus. What has been less widely reported is that those results have substantially been walked back, due to mistakes in error analysis that we can reproduce here. The presence of phosphine would result in a very small dip in the spectrum of Venus, and if the uncertainty in the dip amplitude is incorrect, one can falsely believe random noise is a real signal. The spectrum is given as brightness as a function of frequency. We'll express frequency in km/s relative to the (known) frequency of the phosphine line. The figure shows the published version of the apparently convincing phosphine detection.

(a) One of the challenges in looking for faint dips is that we need to know what the signal would have been in the absence of a dip. To do this, the authors fit a 12th order polynomial (so 13 coefficients) to the data within ± 40 km/s of the phosphine line (so 80 km/s total). For now, treat the phosphine line as a Gaussian with full-width half-max (FWHM) of 12 km/s (as a reminder, FHWM=$\sqrt{(8\ln(2))}\sigma$). For simplicity, we'll keep the width and center of the Gaussian fixed and only fit its amplitude. What is the *ratio* of the errors when you fit only the Gaussian to when you simultaneously fit the Gaussian plus 12th order polynomial? You should be able to do this without actual data. (Technically, the model should be the product of a polynomial and a Gaussian, but for relatively smooth backgrounds, we can get away with treating as a sum.)

Show that you get the same answer when using both Chebyshev and Legendre polynomials (the functions np.polynomial.legendre.legvander and np.polynomial.chebyshev.chebvander may be useful). Show that your answer is only wealy sensitive on how many x points you use, as long as their spacing is much smaller than the width of the dip. To be clear, you can use "x=np.arange(-40,40,n)", and "y=np.exp(-0.5*(x/sigma)**2)" for suitably large n and correctly chosen sigma.

(b) Repeat (a) using a Lorentzian with FWHM 12 km/s (if a Lorentzian is $f(x) = \frac{1}{1+(x/a)^2}$ then the FWHM is $2a$).

(c) The *claimed* detection significance was 15σ where the effects of the background polynomial fitting were not included. Do you believe those claims? Please justify your answer.

Problem 2:

We saw in class that we could stabilize the centered-derivative version of the advection equation by adding what turned out to be artificial numerical viscocity via the Lax technique. In some ways this is not ideal since the viscocity artificially damps our solution. In this problem, we will try to trade off reducing

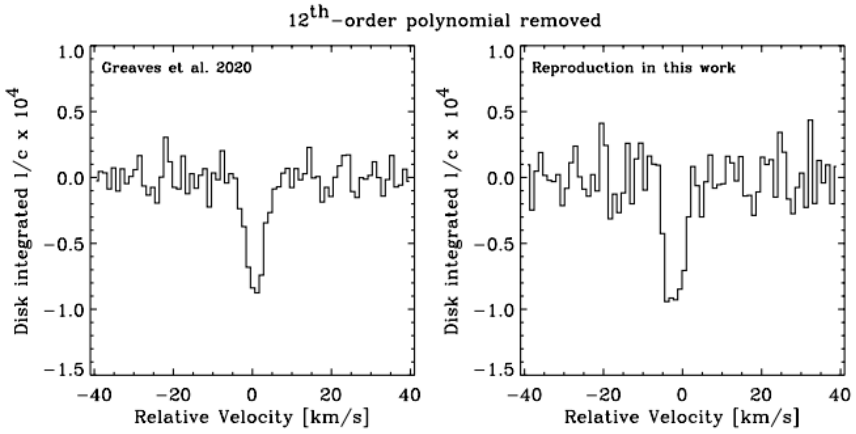

Fig. 1. Reproduction of the ALMA line-spectrum as presented by Greaves et al. 2020, with the original and reproduction in the left and right panel respectively. This is after a 12^{th}-order polynomial is removed from the spectral baseline. The reproduced spectrum is scaled down artificially by a factor 12.8/16.1 to account for the different continuum brightnesses used in the studies. In the reproduction, the line-feature shows a small velocity offset, and the spectral baseline is somewhat more noisy, but the overall signal-to-noise ratio of the two features is similar.

Figure 1: Phosphine data from Snellen *et al.* The left was original published curve, the right from an independent reanalysis. The big question - could the dip be noise?

the viscocity and shorter timesteps while remaining stable.

(a) Recall that in the Lax scheme, we modified the usual discretized advection equation

$$f(x, t + \delta t) = f(x, t) - v\delta t \partial(f(x, t))/\partial x$$

by replacing $f(x, t)$ on the right hand side with $(f(x - \delta x, t) + f(x + \delta x, t))/2$. We'll introduce a parameter β such that that term on the right hand side is $\beta f(x, t) + (1 - \beta)(f(x + \delta x, t) + f(x - \delta x, t))/2$. If $\beta = 1$ we have no numerical viscocity, if we have $\beta = 0$ we're back at the full Lax scheme. Further, let us use the centered derivative $\partial f(x, t)\partial x = (f(x + \delta x, t) - f(x - \delta x, t))/2\delta x$, since it's accurate to second order. Define $\alpha = v\delta t/\delta x$. If we have $f(x, t) = \exp(ikx)$,

what would $f(x, d + \delta t)$ equal in terms of α, β *etc.*? To be clear, we have:

$$f(x,t+\delta t) = \beta f(x,t)+(1-\beta)(f(x+\delta x,t)+f(x-\delta x,t))/2-\alpha(f(x+\delta x,t)-f(x-\delta x,t))/2$$

(b) We saw repeatedly that the shortest wavelengths usually are the first to diverge when an integration scheme is unstable. If we have grid cells spaced by δx, what is the *maximum* value of k our gridded solution can support? You may wish to think back to the Nyquist theorem...

(c) Rather than try to solve the full stability criterion from part (a), set k to be the maximum k from part (b) in your answer to (a). This simplifies matters considerably and you can now write down a closed form constraint on α and β to ensure stability. What is the constraint?

(d) You might think that you could get away with less viscocity if you restrict your starting conditions to have only contributions from small values of k. Explain why this is a bad and dangerous game to play, and why you should continue to use the stability constraint from (c). (β closer to 1)

Problem 3: For the $n-$body problem, one often finds that energy is not conserved because time steps were too large given the softening scale. In this problem we will work out for 2 particles what the relationship between the softening scale and the time step should be.

(a) Let's assume we have particles that are spheres of uniform density, and all have the same mass M, and radius r_0. We first need to work out the potential energy from one of these spheres. If one sphere is centered at $r = 0$, what is the gravitational energy released when we bring the second sphere from infinity to $r = 0$? (a possibly useful reminder is that the potential energy in a uniform sphere is $-\frac{3}{5}GM^2/R$ (5 marks)

(b) If I have two particles that start at rest at infinity, what is the *maximum* velocity each particle can attain? (3 marks) Our solver is should be well behaved if the time step is much smaller than the shortest time during which the particles pass each other. What is this time? (2 marks)

(c) What is a gravitational force law you could use that would have the right behavior when the spheres do not overlap, and that has the right limiting energy when they overlap perfectly? The exact force law from spheres is complicated, but you can approximate it with a force that is linear in r when the spheres overlap, is exact when they don't, and is continuous everywhere.

(d) Write a 1-d python code called "nbody_pair.py" that starts two equal-mass particles at rest separated by at least 20 times the particle radius and evolves it with time, using your force law from part (c). Show that for sufficiently small timesteps, your max velocity agrees with what you expect.

Problem 4:

You may have seen stories claiming evidence that exposure to radio waves can cause cancer. In this problem, we will use data from the US National Toxicology Program study to decide if you should throw away your cell phone

or not. The program "cancer_like.py" has the data and will calculate the log likelihood for various types of cancer.

By way of background, the study exposed a set of rats to varying intensities of radio waves and looked at the incidence of gliomas (a type of brain cancer) and schwannomas (a type of heart cancer). They claim that for male rats, increasing intensity of radio frequency (RF) radiation was correlated with increasing rates of cancer. To test this, we will fit a model to the data that the cancer rate is equal to a constant background rate plus another constant time the exposure level, $p_{cancer} = p_{background} + \alpha I$ where p_{cancer} is the expected probability of getting cancer, $p_{background}$ is the probability of getting cancer in the absence of any RF exposure, I is the RF exposure level, and α is the relation between the exposure level and the cancer rate. If we can confidently state that $\alpha > 0$, then increased exposure leads to increased cancer rates, and you should be careful with your phones. Clearly, knowing the background incidence rate is important, since we are looking for an excess number of cancers over that rate. This study present control data on 90 rats who were not exposed to radiation, but also include background rates from the literature from other experiments. We will also look at the effects of including the extra control data.

If you look inside cancer_like.py, you'll see I have already written much of the code you'll need. It evaluates the likelihood using Poisson statistics, which is what you use when counting discrete events (like the number of cases of cancer). The data from the paper are in the main part of the code. In all cases, 90 rats each were tested for no radiation (the control sample), then 1, 2, and 4 times some nominal dose for two different kinds of cell phone signals (GSM, used broadly around the world, and CDMA, used widely in North America). The likelihood takes a two-element list/array; the first is the background rate, the second is the change in rate with exposure.

(a): Write a routine to create a Markov chain for glioma data. Be careful: the likelihood routine returns the log of the likelihood, *not* χ^2. Your routine should reflect this. After you run, justify why you think your chain is converged. What fraction of your samples have $\alpha < 0$?

Note: If you can't get your MCMC code to work, you can use pre-calculated chains included in the exam repository. Than can be read with *e.g.* "chain=np.load('chain_glioma.npy')".

(b): Run your code from (a), but using the schwannoma (a type of heart cancer) data.

(c): One of the potential issues with this study is that the control sample had zero cancers. If this is a statistical anomaly, then the inferred effects from RF would be overestimated. One possibility is to include control samples from other studies to improve the background rate estimate. Control data from other, much larger, studies is also included in the files (look for things with "_wold" in the tag/chain name). What do you estimate for the statistical significant of these cancers when including the old control data? What could go wrong when including control groups from other studies?

(d): The study also reported rates for several other cancer types. How would

this affect your interpretation of results for schwannomas and gliomas? The study also failed to find any incidence of schwannomas/gliomas in female rats, or in mice. Now how do your feel about the reported statistical significance?

(e): Another potential issue in this study is that the number of rats in each group that survived at the end of two years were [25,45/43,50/56,60/43] for the control, low, medium, and high-dose male samples. The two results quoted for the non-zero exposure were for GSM/CDMA exposure. How could these numbers (qualitatively) affect your interpretation of your chain results?

I hope this problem has given you a sense for how tricky it can be when looking for many possible effects in a small sample. You might ask yourself how surprising it is that half of published medical results are not replicated. You might also ask yourself how nervous you feel about talking on your phone. BTW, the low-exposure category corrsponded to using a phone at the maximum recommended power limit, of 1.5 W/kg absorbed by tissues, for 18 hours a day. For reference, phones usually broadcast a *total* RF power of 0.25W these days.

Problem 5:

In this problem, we will work out when it is faster to use Fourier-based techniques in $n-$body simulations instead of directly summing forces between each pair of particles.

(a) First, we will work out how long it will take to calculate gravity by Fourier transforming a grid. If we have a 3-dimensional grid with m cells along each side, how many total cells do we have?

(b) Roughly how many operations will we need to carry out to do the Fourier transform? I don't care about the coefficient, but I do care about the scaling with m. We will assume (usually justifiably so) that the work for the mesh is dominated by the FFTs. Let the run time equal a times this operation count.

(c) If we instead calculate the forces between every pair of particles, how many operations do we need to do for n particles? Again, the scaling with n needs to be correct but don't sweat the coefficient. Let the run time equal b times the operation count (so, if the operation count scaled like n to the first power, our run time would be bn.)

(d) Given the scalings from the previous questions, what is the critical value for m at which the grid-based and brute-force techniques take the same run time? In practice, though hard to predict, the coefficients a and b will be of similar magnitude, and easy to measure with quick timing runs. Note - if you get a transcendental equation, feel free to treat the logarithm of m as a constant. In practice, you can pick a starting value for the log, solve for m keeping that fixed, then repeat using your updated values of m in the log. This converges extremely quickly.

(e) Interpret your answer to part (d) and write down a rule of thumb for when a grid will be faster than brute-force in three dimensions. For instance, one possible answer would be "a grid will be faster when there are many particles in each 3-dimensional grid cell, and slower when the average number of particles per cell is much less than one." This may or may not be correct, but your

answer should look something like that. There are many possible answers, but one useful one is to express the average number of particles per cell in terms of the total number of particles plus numerical constants.