Gas Radial Profile Fitting

Shangjia Zhang, Zhaohuan Zhu

June 1, 2018

Abstract

We use piece-wise function in the form of cos(tanh(x)) to fit the gaps in gas radial profile. We summarize fitting parameters for all our runs in a table and provide a python tool to read and plot the curves.

1 Data

We do a parameter study of proplanetary disks using FARGO 2D, with a combination of h/r = 0.05, 0.1; alpha = 1e-4, 1e-3, 1e-2; and planet mass = 3.3e-5, 1e-4, 3.3e-4, 1e-4, 3.3e-4, 1e-3 in code unit, 30 runs in total. The code unit mass is in solar mass and code unit length is 20 AU. The planet is at r = 1. The initial density profile is 0.00045 $\times$ r^{-1} in code unit. We fit all 30 runs at 1000th orbit and the runs with viscosity equals to 1e-4 and 1e-3 at 100th orbit, to see the whether their evolution reach steady stage.

2 Fitting

First, azimuthal averaged gas density is divided by the initial condition.

$$\Sigma_0(r) = Ar^p \tag{1}$$

where A = 4.5e-4 and p = -1.

$$\sigma(r) = \Sigma(r)/\Sigma_0(r) \tag{2}$$

We then fit the normalized density profile in logarithmic scale and r in linear scale ($log_{10}\big(\sigma(r)\big) - r$), using least square fitting. First, the local maxima and minima are found on the curve and then a modified half-cosine function were fitted to connect these extrema. This modified half-cosine function is a variation from Kees Dullemond's piece-wise cosine function in DISKLAB, but stretched near 0 or 2π and squeezed around π. The purpose is to better fit the flat shape around maxima and minima and the rapid change of the flank together. Take r_i and r_j for the left and right positions of the neighboring extrema, coordinates r in between are normalized to a number between 0 and 1. And we use tanh(x) to modify the changing rate from 0 to 1.

$$y = (r - r_i)/(r_j - r_i) \tag{3}$$

where y ranges from 0 to 1 and is a linear mapping of r. Then the fitting gives us two points on tanh(x) function, x_i and x_j. We confine $x_i < 0$ and $x_j > 0$. The region between x_i and x_j (after normalization) becomes a new mapping from [0, 1] to [0, 1], but it is no longer linear. (see plots in Figure 1.).

$$x = y(x_j - x_i) + x_i = \frac{r - r_i}{r_j - r_i}(x_j - x_i) + x_i \tag{4}$$

$$\mu(r; x_i, x_j) = \frac{tanh(x) - tanh(x_i)}{tanh(x_j) - tanh(x_i)} \tag{5}$$

Now $\mu(r)$ still ranges from 0 to 1, but the slope is smaller around 0 and 1 and larger in the middle, comparing to the original half-cosine function (See Figure 1.). x_i and x_j are fitting parameters here. The fitting function is:

$$f(r) = f(r_i) + 0.5\big(f(r_i) - f(r_j)\big)\big(cos(\pi\mu(r; x_i, x_j)) - 1\big) \tag{6}$$

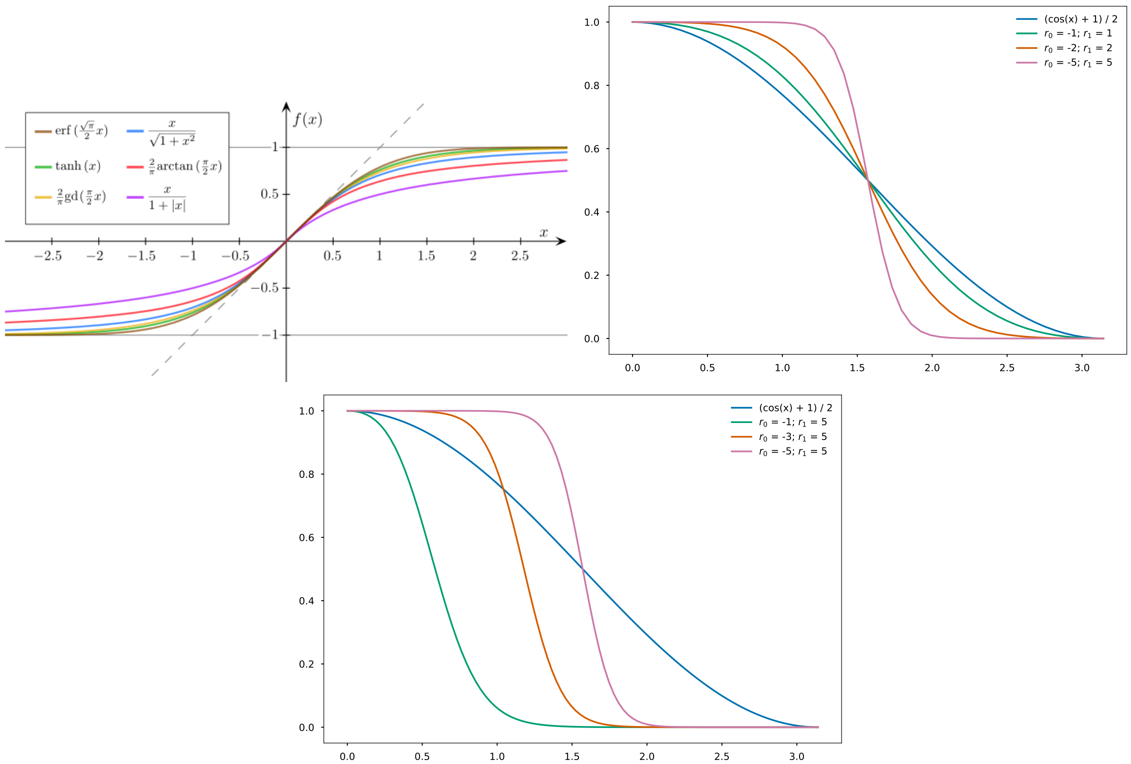

Figure 1

To construct the function back from parameters in the table, we have

$$f(r) = \begin{cases} f(r_i) + 0.5\big(f(r_i) - f(r_j)\big)\big(cos(\pi\mu(r; x_0, x_1)) - 1\big) & r_i \leq r < r_j \\ 0 & elsewhere \end{cases}$$

and

$$F(r) = \sum_{k=1}^{n} f_k(r) \tag{7}$$

Now F(r) is $log_{10}\Big(\Sigma(r)/\Sigma_0(r)\Big)$

The normalized density profile is

$$\sigma(r; r_i, r_j, f(r_i), f(r_j), x_i, x_j) = 10^{F(r; r_i, r_j, f(r_i), f(r_j), x_i, x_j)} \tag{8}$$

If there are n pieces (half-cosine), we have $6n$ parameters ($4n$ is from the coordinates of the extrema and $2n$ is from the fitting). But since neighboring pieces share the same point, the number of free parameters is $4n$+2.

Finally, the original density profile is

$$\Sigma(r; A, p, r_i, r_j, f(r_i), f(r_j), x_i, x_j) = \Sigma_0(r; A, p) \times 10^{F(r; r_i, r_j, f(r_i), f(r_j), x_i, x_j)} \tag{9}$$

where A = 4.5e-4 and p = -1 in our case.

3 Table

The fitting parameters are summerized in a table. A csv version of the table *fitting_params.csv* is in the folder and a pdf version table containing parameters for 1000th orbit is named *table*1000.*pdf*.

4 Tool

A python tool with the functions from reading the table to plotting curves are in *read_table.py*. It requires numpy, pandas and matplotlib as three basic packages. A jupyter notebook with a short tutorial is also provided. Docstring in the functions contain help information.

Please email shjzhang@umich.edu if you meet any problem.