



*Mapana Journal of Sciences*  
2019, Vol. 18, No. 1, 67-92  
ISSN 0975-3303 | <https://doi.org/10.12723/mjs.48.5>

# On Self-Gravitating Polytropic Cylinders and Slabs

Mandyam N Anandaram\*

## Abstract

In this review paper the 2-D Lane-Emden equation (LEEq) model of a self-gravitating gas distribution in the form of an infinitely long cylinder shaped polytrope of finite radius is obtained and its basic radial properties are outlined. Similarly reviewed is the derivation of the 1-D LEEq model of an infinitely wide planar polytrope of finite thickness and its basic properties across thickness are discussed. These two polytropes are solved numerically along with the 3-D models for comparison using the 2<sup>nd</sup> order Euler-Richardson method (ERM) and their index based parameters are determined. The Python script used in these computations has been shown to be not only fast but is capable of matching fourth order performance. However, these models are found to have finite radii for all polytropic indices unlike the restricted spherical analogs and have astrophysical applications. Distortion due to rotation in polytropic rings has also been computed using ERM.

**Keywords:** Polytropic Cylinders, Lane-Emden Models, Euler-Richardson Solutions

## 1. Introduction

Huge self-gravitating gas distributions have been detected over many decades in many parts of the galaxy such as ISM, molecular clouds and even spiral arms. The approximate shape of these structures is in the form of huge planar sheets and long cylindrical

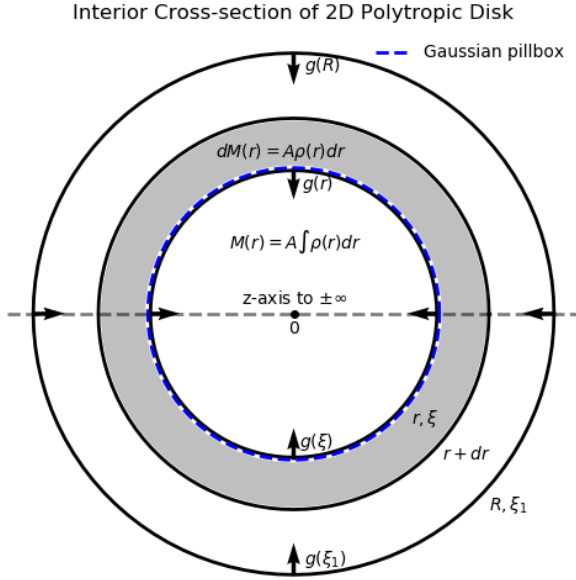
---

\* Bangalore University, Bangalore, India; [mnanandaram@gmail.com](mailto:mnanandaram@gmail.com)

filaments. Idealised models of such gas condensations consist of treating them as infinitely sized planar polytropes of finite thickness or as infinitely long cylindrical polytropes of finite radii respectively. While these polytropes have a reduced dimensionality (1-D and 2-D respectively), their theory is similar to that of spherical polytropes described by the Lane-Emden equation (LEE<sub>q</sub>, [1]). The theory and properties of simple polytropic cylinders will be briefly reviewed below followed by polytropic slabs as both the polytropes find applications for astrophysical modelling in many publications.

### 2.1 The 2-D Lane-Emden Equation of Cylindrical Polytropes

Consider an infinitely long cylinder of finite radius containing polytropic gas (or fluid) in a state of self-gravitating hydrostatic equilibrium. The problem is to determine the run of physical variables which have only a radial dependence from the axis of the cylinder. In this sense, it is a 2-D problem and so we may consider a thin disc of radius  $R = \alpha \xi_1$ , axial length  $dz$  and linear mass  $M$  in the cylinder. If we consider a cylindrical ring of radius  $r$ , width  $dr$  and mass  $dM(r)$  then it is gravitationally pulled by the mass  $M(r)$  of the inner disk toward the axis. To find this force, a Gaussian pillbox is placed around the inner disk of radius  $r$  and mass  $M(r)$ . If  $g(r)$  denotes the radial gravitational field normal to the curved side which has a surface area  $dA = 2\pi r dr$ , the total gravitational flux is  $g(r)dA$ . Then, the Gauss theorem requires that  $g(r)dA = 4\pi GM(r)$ . Hence,  $g(r) = 2GM(r)/r$  is the inward acceleration per unit mass of the surrounding ring which must be balanced by the outward radial force  $dP(r)dA$ . If  $\rho(r)$  denotes the density, we can write  $dP(r)dA = -g(r)\rho dr dA$ , or,  $dP(r)/dr = -g(r)\rho = -2GM(r)\rho/r$ . This can be rewritten as  $(r/\rho) dP(r)/dr = -2GM(r) = 2G \int_0^r 2\pi r \rho dr$ . This expression is differentiated again to remove the integral which, after simplification, reduces to the 2-D Poisson equation given by



**Figure.2.1:** Shown here is the interior cross-section of a disk of the radius  $\mathbf{R} = \alpha \xi_1$  (and height  $dz$ ) along with a Gaussian pillbox covering a mass  $M(r)$  for calculating the acceleration  $g(r)$  due to it on a unit mass placed at the same distance.

$$\frac{1}{r} \frac{d}{dr} \left( \frac{r}{\rho} \frac{dP}{dr} \right) = -4\pi G \rho \tag{2.1}$$

Here the radial parameter has one degree less than that in the 3-D Poisson equation obtained for spherical polytropes [1,2]. Both can be combined into a single expression by rewriting (2.1) as

$$\frac{1}{r^{(D-1)}} \frac{d}{dr} \left( \frac{r^{(D-1)}}{\rho} \frac{dP}{dr} \right) = -4\pi G \rho \tag{2.2}$$

Here  $D$  denotes the dimensionality of the model problem and has values 2 and 3 for the 2-D Poisson and the 3-D Poisson equations respectively. It can be expected that the value  $\mathbf{D} = \mathbf{1}$  would correspond to a 1-D Poisson equation and this will be proved to be so in Sec.3. We can now obtain the 2-D Lane-Emden equation (LEE) in the same way as was done in [1] for spherical polytropes of polytropic index  $n$  by substituting the polytropic EOS,  $\mathbf{P} = \mathbf{K} \rho^\gamma$  where  $\gamma = \mathbf{1} + \mathbf{1}/n$  into (2.1) to get

$$\gamma \frac{K}{r} \frac{d}{dr} \left( r \rho^{\gamma-2} \frac{d\rho}{dr} \right) = -4\pi G \rho \tag{2.3}$$

Now, as in [1] define  $\rho(r) \equiv \rho_c \theta^n$  where  $\rho_c$  is the central (=axial) density,  $n$  is the polytropic index and  $\theta \equiv \theta(r)$  is the parameter that fixes the density, we require  $\theta = 1$  along the axis since  $\rho = \rho_c$  there and  $\theta = 0$  since  $\rho = 0$  at the circular surface of the cylinder. Using this definition in (2.2) and simplifying it along with,

$$r \equiv \alpha \xi \equiv \left[ (n + 1)K / \left( 4\pi G \rho_c^{1-1/n} \right) \right] \xi, \tag{2.4}$$

where  $\xi$  is the scaled radius of the disc, we get the 2-D LEEq of the cylindrical polytrope noting that this is **different** from the similar 3-D LEEq ([1], [2]) as

$$\frac{1}{\xi} \frac{d}{d\xi} \left( \xi \frac{d\theta}{d\xi} \right) = \theta'' + \frac{\theta'}{\xi} = -\theta^n. \tag{2.5}$$

Here  $\theta \equiv \theta(\xi) \equiv \theta_n(\xi)$  is the Emden function for the cylindrical polytrope of index  $n$  as well as the solution,  $\theta' = d\theta/d\xi$  and  $\theta'' = d^2\theta/d\xi^2$ . The generalised LEEq can now be obtained from (2.2) and the expressions of (2.5) become generalised[7] as

$$\frac{1}{\xi^{(D-2)}} \frac{d}{d\xi} \left( \xi^{(D-1)} \frac{d\theta}{d\xi} \right) = \theta'' + (D - 1) \frac{\theta'}{\xi} = -\theta^n \tag{2.6}$$

Here too, the same values ( $D = 3, 2, 1$ ) and comments which followed (2.2) apply.

The first five terms of the power series solution of the 2-D Emden function are found [3] to be:

$$\theta(\xi) = 1 - \left(\frac{1}{4}\right)\xi^2 + \left(\frac{n}{64}\right)\xi^4 - \left(\frac{n(3n-2)}{2304}\right)\xi^6 + \left(\frac{n(18n^2-29n+12)}{147456}\right)\xi^8 \mp \dots \tag{2.7}$$

The radius of the cylindrical polytrope is defined as  $\xi_1$  at which  $\theta(\xi_1) = 0$ . For  $n = 0$  this is found from (2.7) to be  $\xi_1 = 2$  and for  $n = 1$ , it is found that (2.7) reduces to the Bessel function  $J_0(\xi)$  so that its first root lies at  $\xi_1 = 2.404825558$ . In the next section,  $\theta(\xi)$  and  $\theta'(\xi)$  will be solved numerically in a computer for a few values of the polytropic index ( $0 \leq n \leq \infty$ ). Now the first three terms in the derivatives of (2.6) needed for computer solutions are as follows:

$$\theta' = -\xi/2 + \left(\frac{n}{16}\right)\xi^3 - \left(\frac{n(3n-2)}{384}\right)\xi^5 \pm \dots \quad (2.8)$$

$$\theta'' = -\frac{1}{2} + \left(\frac{3n}{16}\right)\xi^2 - \left(\frac{5n(3n-2)}{384}\right)\xi^4 \pm \dots \quad (2.9)$$

The mass and mean density will now be calculated. The mass per unit length within a radius  $r$  is given by (2.10) and after integrating, it takes the simplified form shown in (2.11) below.

$$M(r) = 2\pi \int_0^r \rho(r)rdr = M(\xi) = 2\pi\rho_c\alpha^2 \int_0^\xi \theta^n \xi' d\xi' \quad (2.10)$$

$$M(\xi) = 2\pi\rho_c\alpha^2 \int_0^\xi [d(-\xi'\theta')/d\xi'] d\xi' = 2\pi\rho_c\alpha^2 |\xi\theta'| \quad (2.11)$$

The quantity  $|\xi\theta'|$  known as the mass per unit length parameter is the absolute value of the product of radius and the slope  $\theta'$  at that interior point. Its maximum value  $|\xi\theta'|_{\xi_1}$  occurs at the edge of the polytropic disk where  $\xi = \xi_1$  and this value is listed in Table 2.1. Thus, the total mass (per unit length) of the disk is  $M = 2\pi\rho_c\alpha^2 |\xi\theta'|_{\xi_1}$ . As the numerical values of  $\theta(\xi)$  and  $\theta'(\xi)$  are computed, the related physical parameters such as the density fraction,  $\rho(\xi)/\rho_c = \theta^n$ , the pressure fraction,  $P(\xi)/P_c = \theta^{n+1}$ , the product  $\xi\theta'$  are quickly computed and finally, the mass fraction  $M(\xi)/M = |\xi\theta'|/|\xi\theta'|_{\xi_1}$  is also computed.

The volume per unit length of the disk within a radius  $r = \alpha\xi$  is  $V(\xi) = \pi\alpha^2\xi^2$  so that the mean density interior to this radius is  $\bar{\rho}(\xi) = M(\xi)/V(\xi) = 2\rho_c|\theta'/\xi|$ . When this expression is evaluated over the entire disk, we get  $\bar{\rho} = \rho_c|2\theta'/\xi|_{\xi_1}$ . Hence, after inverting it, the central condensation is given by the ratio,

$$\rho_c/\bar{\rho} = |\xi/2\theta'|_{\xi_1} \quad (2.12)$$

The magnitude of central condensation is also computed for selected polytropic indices and is shown in the last column of Table 2.2.1.

### 2.2. Solving the 2-D Polytropic Cylinder Model

The computer solution of the corresponding 2-D LEEq can be carried out by first writing (2.5) as a coupled set of two first order ODEs using two new variables  $y$  and  $z$  defined by  $y \equiv \theta$  and  $z \equiv y' = d\theta/d\xi$  as

$$dy/d\xi = z ; dz/d\xi = -(z/\xi) - y^n \tag{2.13}$$

with the BCs now reading as  $y(\xi = 0) = 1, z(\xi = 0) = 0 ; y(\xi_1) = 0, z(\xi_1) < 0$ . Two simple but rapid methods of integrating (2.13) were considered and tested here. They are respectively the stable first order Euler-Cromer Algorithm (ECM, [6]) and the second order Euler-Richardson Algorithm (ERM, [7]) both of which are discussed in Appendix A. Trials were conducted first with ECA and a step size like  $\Delta\xi = 0.00002$  to determine four parameters for several polytropic indices. They were then compared with a more accurate computer solution based on the use of the SciPy integrator module ‘dopri5’ as it was capable of fifth order error controlled adaptive integration (see [2] for details). The parameters are listed in Table 2.2.1 below.

**Table 2.2.1:** 2-D Polytropic Cylinder Radial Parameters

Computing Algorithm: BVP with fitting + dopri5(*) integrator (see Ref: 2)				
Index	Emden Radius	Slope at $\xi_1$	Mass param.	Axial Rel. Dens.
N	$\xi_1$	$ \theta' _{\xi_1}$	$M(\xi_1) \propto  \xi\theta' _{\xi_1}$	$\rho_c/\bar{\rho} =  \xi/2\theta' _{\xi_1}$
0.0	2.0	-1.0	2.0	1.0
1.0	2.40482555770	-0.51914749729	1.24845916969	2.316129395
1.5	2.64777676622	-0.40075689248	1.06111478882	3.303470029
3.0	3.57390098193	-0.20709082996	0.74012212054	8.628824808
5.0	5.42757458624	-0.09808145384	0.53234440623	27.668709903
8.0	10.4029169860	-0.03622778865	0.37687467788	143.576483339
10.0	16.2227407494	-0.01947836994	0.31599254581	416.429629293
12.0	25.4531321698	-0.01069534195	0.27222995235	1189.916707644

It was found that the ECM method produced values within 0.001% of the values seen in the table. Then, the trial with the larger step size,  $\Delta\xi = 0.00005$ , was repeated with ERM and it was found that the ERM-produced values were within a remarkable 0.000001% from the values in Table 2.2.1. Hence, only these parameter values

are reported in Table 2.2.2. They are respectively the Emden radius in the 2<sup>nd</sup> column, the slope of Emden function in the 3<sup>rd</sup> column, mass per unit length in the 4<sup>th</sup> column and the relative axial density (central condensation) in the last column against the polytropic index in the first column. These parameters are also in excellent agreement with similar values reported by Horedt in Table 2.5.2 of [9].

In addition, the values listed in Table 2.2.2 are broadly in agreement but are more accurate than the values computed up to 6 digits over 53 years ago by Ostriker ([3] and [4]) using the standard Runge-Kutta algorithm. A perusal of the values in Table 2.2.2 indicates that as  $n$  increases the radius (2<sup>nd</sup> col) of the polytropic cylinder gets larger. Therefore, the mass parameter decreases (3<sup>rd</sup> col). This implies that the axial region gets more massive and this is confirmed by the rapid increase of central condensation along the axis as shown by the listing in the last column.

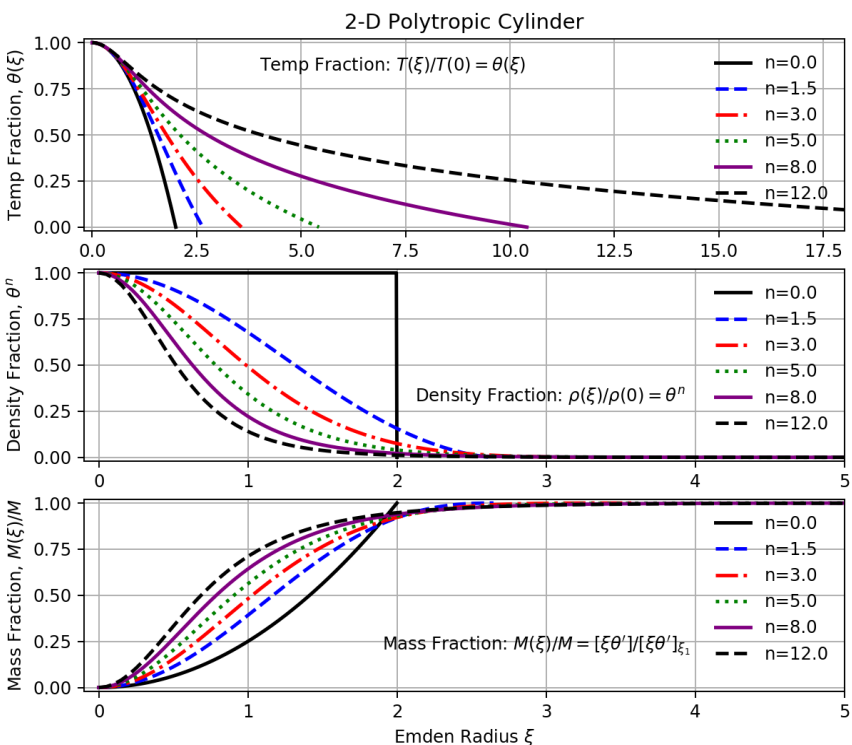
**Table 2.2.2:** 2-D Polytropic Cylinder Radial Parameters

Computing Algorithm: Euler-Richardson with step size $\Delta\xi = 0.00005$				
Index	Emden Radius	Slope at $\xi_1$	Massparam.	Axial Rel. Dens.
n	$\xi_1$	$ \theta' _{\xi_1}$	$M(\xi_1) \propto  \xi\theta' _{\xi_1}$	$\rho_c/\bar{\rho} =  \xi/2\theta' _{\xi_1}$
0.0	2.0000000000	-1.0000000000	2.0000000000	1.0000000000
0.5	2.1896621900	-0.6975388627	1.5273744737	1.569562864
1.0	2.4048255572	-0.5191474973	1.2484591694	2.316129395
1.5	2.6477767660	-0.4007568923	1.0611147881	3.303470030
2.0	2.9213207237	-0.3167585341	0.9253532702	4.6112739024
3.0	3.5739009830	-0.2070908298	0.7401221201	8.628824818
5.0	5.4275745908	-0.0980814537	0.5323444059	27.668710164
8.0	10.4029170047	-0.0362277886	0.3768746776	143.576483968
10.0	16.2227407555	-0.0194783699	0.3159925458	416.429629681
12.0	25.4531322552	-0.0106953419	0.2722299521	1189.916716724
20.0	160.5964733254	-0.0010933470	0.1755876774	73442.588987976

In Fig. 2.2., the Radial properties of 2-D Polytropic Cylinders are graphed for  $n = 0, 1.5, 3, 5, 8$  and  $12$ . In the graph, at the top  $\theta(\xi)$  is drawn against  $\xi$  which increases with  $n$ . In the middle graph, the density fraction  $\theta^n$  is drawn magnified against  $\xi$  showing that it increases with  $n$  and moves closer to the central axis of the polytrope. The constant density profile of the  $n = 0$  polytropes may

also be noted. In the graph at the bottom, the mass fraction (normalised mass distribution) is drawn against  $\xi$  showing that more mass collects closer to the central axis due to an increase in the density with index  $n$  (see Table 2.2.2). Thus, as  $n$  increases to large values, the central condensation also rapidly increases while the polytropic disk becomes extended. However, the radius always remains finite (unlike that of polytropic spheres).

In this way, the computational work done here with the second order ERM has enabled solving the 2-D LEEq and computing quite accurate values of the parameters for several indices in fast sequence. Among the many applications of the cylindrical polytrope model, a recent one is described in [9].



**Figure 2.2.1:** Radial parameter variations of 2-D Polytropic Cylinders are shown graphed together in three groups for  $n = 0, 1.5, 3, 5, 8$  and  $12$ . At the top, middle and bottom the temperature fraction  $\theta(\xi)$ , the density fraction  $\theta^n$  and the mass fraction are respectively drawn against  $\xi$  (see text for details).



The problem of computing the extent of distortion due to rotation in 2-D polytropic rings [4] has been briefly dealt with in Appendix B as an example application of the ERM described above and all the computed parameters listed in Table B.1 as several polytropic indices are in very good agreement with those given in [4].

### 3.1 The 1-D Lane-Emden Equation of Planar Polytropes

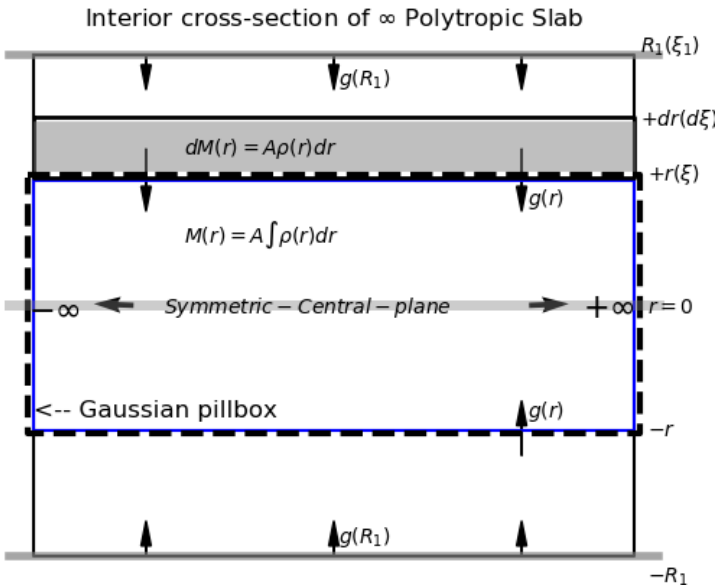
Consider an infinitely large planar slab of finite thickness containing polytropic gas (or fluid) in a state of self-gravitating hydrostatic equilibrium with respect to the flat central plane of symmetry in the middle of the slab. The problem then is to determine the run of physical variables which have only height dependence from this central plane on both sides and in this sense, constitutes a 1-D problem. So, as shown in **Figure 3.1**, we may consider a slab of height  $R_1 = \alpha \xi_1$  both above and below the **central plane**, a total thickness  $2R_1$  which equals the slab thickness and a total mass density  $M$  per unit area in the slab. If we consider a cylindrical ring of horizontal radius  $a$ , height  $dr$  and mass  $dM(r) = 2\rho(r)dr$ , it is gravitationally pulled by the mass  $M(r)$  of the inner disk toward the central plane. To find this force, a Gaussian pillbox is placed around the inner disk of horizontal radius, *thickness*  $2r$  and mass  $M(r)$ . If  $g(r)$  denotes the vertical gravitational field normal to the upper flat face which has a surface area  $dA = \pi a^2$ , the gravitational flux through it is  $g(r)dA$ . **An equal flux contribution comes from the lower flat face so this makes the total flux through the pillbox to be  $2g(r)dA$  directed at the central plane.** Gauss theorem now specifies that  $2g(r)dA = 4\pi GM(r)dA$ . Hence  $g(r) = 2\pi GM(r)$  is the **inward** acceleration acting on a unit mass at a height  $r$  from the central plane. The resulting force  $g(r)dM(r)dA$  must then be balanced by the net **outward** vertical force  $dP(r)dA$ . If  $\rho(r)$  denotes the density, we can write,

$$dP(r)dA = -g(r)\rho drdA,$$

$$dP(r)/dr = -g(r)\rho = -2\pi GM(r)\rho. \tag{3.1}$$

This can be rewritten as

$$\left(\frac{1}{\rho}\right) \frac{dP(r)}{dr} = -2\pi GM(r) = -2\pi G \int_0^z 2\rho(r) dr.$$



**Figure 3.1:** Shown here is the interior cross-section of a cylindrical disk (having a flat surface area  $dA$ ) cut from an infinite polytropic slab along with the Gaussian pillbox surrounding a mass  $M(r)$  to find its gravity  $g(r)$  at both heights  $+r$  and  $-r$ .

It may be noted here that  $M(r)$  is the sum of masses (per unit area) enclosed by the pillbox both above and below the central plane. This expression is differentiated again and after inserting  $\frac{dM(r)}{dr} = 2\rho(r)$ , may be rewritten as

$$\frac{d}{dr} \left( \frac{1}{\rho} \frac{dP}{dr} \right) = -2\pi G \frac{dM(r)}{dr} = -4\pi G \rho$$

Thus, we arrive at the 1-D Poisson equation given by,

$$\frac{d}{dr} \left( \frac{1}{\rho} \frac{dP}{dr} \right) = -4\pi G \rho \tag{3.2}$$

This expression results also by inserting the value  $D = 1$  into the generalised Poisson equation (2.2). We can now transform (3.2) to obtain the 1-D planar LEEq in the same way as was done in Sec.2

for cylindrical polytropes of index  $n$  by inserting the EOS,  $P = K\rho^\gamma$  into (3.2) to get

$$\gamma K \frac{d}{dr} \left( \rho^{\gamma-2} \frac{d\rho}{dr} \right) = -4\pi G\rho \quad (3.3)$$

Defining  $\rho(r) \equiv \rho_c \theta^n$  where  $\rho_c$  is the density at the central plane,  $n$  is the polytropic index and  $\theta \equiv \theta(r)$  is the (temperature) parameter that fixes the density. We require that  $\theta = 1$  along the central plane since  $\rho = \rho_c$  there and also  $\theta = 0$  since  $\rho = 0$  at both the equidistant flat surfaces of the slab. Inserting this definition in (3.2) and simplifying it using the same definition  $r \equiv \alpha\xi$  as given in (2.4) where the scaled height from the central plane is now  $\xi$ , we get the 1-D LEEq of the polytropic slab as

$$\frac{d}{d\xi} \left( \frac{d\theta}{d\xi} \right) = \theta'' = -\theta^n. \quad (3.4)$$

Here  $\theta \equiv \theta(\xi) \equiv \theta_n(\xi)$  is the 1-D Emden function for the polytropic slab of index  $n$  as well as the solution,  $\theta' = d\theta/d\xi$  and  $\theta'' = d^2\theta/d\xi^2$ . This expression also results by inserting  $D = 1$  into the generalised Lane-Emden equation (2.6). The first five terms of the usual power series solution of this equation were determined using standard procedure [1], to be

$$\theta(\xi) = 1 - \left(\frac{1}{2}\right)\xi^2 + \left(\frac{n}{24}\right)\xi^4 - \left(\frac{n(4n-3)}{720}\right)\xi^6 + \left(\frac{n(34n^2-63n+30)}{40320}\right)\xi^8 \mp \dots \quad (3.5)$$

The height of either flat surface from the central plane of the polytropic slab is defined as  $\xi_1$  for which  $\theta(\xi_1) = 0$ . For  $n = 0$ , (3.5) shows that  $\xi_1 = \sqrt{2}$  and for all other values of  $n$ , no analytic solution is available. In the next section,  $\theta(\xi)$  and  $\theta'(\xi)$  will be solved numerically in a computer for a few values of the polytropic index ( $0 \leq n \leq \infty$ ). Now the first four terms in the derivatives of (3.5) needed for computer solutions are as follows.

$$\theta' = -\xi + \left(\frac{n}{6}\right)\xi^3 - \left(\frac{n(4n-3)}{120}\right)\xi^5 + \left(\frac{n(34n^2-63n+30)}{5040}\right)\xi^7 \pm \dots \quad (3.6)$$

$$\theta'' = -1 + \left(\frac{n}{2}\right)\xi^2 - \left(\frac{n(4n-3)}{24}\right)\xi^4 + \left(\frac{n(34n^2-63n+30)}{720}\right)\xi^6 \pm \dots \quad (3.7)$$

The mass and mean density will now be calculated. The mass per unit area within a radius  $r$  is given by (3.8) and after integration, it takes the simplified form shown in (3.9) below.

$$M(r) = 2 \int_0^r \rho(r) dr = M(\xi) = 2\rho_c \alpha \int_0^\xi \theta^n d\xi' \tag{3.8}$$

$$M(\xi) = 2\rho_c \alpha \int_0^\xi [d(-\theta')/d\xi'] d\xi' = 2\rho_c \alpha |-\theta'| \tag{3.9}$$

The quantity  $|-\theta'|$  known as the mass per unit area parameter is the absolute value of the slope  $\theta'$  at that interior point. Its maximum value  $|-\theta'|_{\xi_1}$  occurs at both the flat surfaces (edge) of the slab where  $\xi = \xi_1$  (aka, Emden height) and this value is listed in Table 3.2. Thus, total mass (per unit area) of the slab is  $M = 2\rho_c \alpha |-\theta'|_{\xi_1}$ . As the numerical values of  $\theta(\xi)$  and  $\theta'(\xi)$  are computed, other parameters such as the density fraction,  $\rho(\xi)/\rho_c = \theta^n$ , the pressure fraction,  $P(\xi)/P_c = \theta^{n+1}$  are quickly computed and then finally the mass fraction,  $M(\xi)/M = |-\theta'|/|-\theta'|_{\xi_1}$  is also computed.

The volume per unit area of the slab within a central planar distance  $r = \alpha\xi$  is  $V(\xi) = 2\alpha\xi$  so that the mean density interior to this height is  $\bar{\rho}(\xi) = M(\xi)/V(\xi) = \rho_c |-\theta'|/\xi$ . Then the mean density of the entire slab is given by  $\bar{\rho} = \rho_c |-\theta'|/\xi|_{\xi_1}$  and after inverting, the central condensation (central planar density relative to mean density) is given by the ratio,

$$\rho_c/\bar{\rho} = |\xi|/|-\theta'|_{\xi_1} \tag{3.11}$$

The magnitude of central condensation is now computed for selected polytropic indices and is then listed in the last column of Table 3.2.1 below.

### 3.2. Solving the 1-D Planar Polytropic (Slab) Model

The computer solution of the corresponding planar LEEq has been carried out by using the same second-order Euler-Richards on integration method as explained and was used earlier in Section 2.2 with a small step size. The expression in (3.4) is written as a coupled set of two first order ODEs using two new variables  $y$  and  $z$  defined by  $y \equiv \theta$  and  $z \equiv y' = d\theta/d\xi$  as

$$dy/d\xi = z ; dz/d\xi = -y^n \tag{3.12a}$$

The central plane ( $\xi = 0$ ) and both flat surface ( $\pm\xi_1$ ) boundary conditions are given by:

$$y(\xi = 0) = 1, z(\xi = 0) = 0 ; y(\pm\xi_1) = 0, z(\pm\xi_1) < 0 \tag{3.12b}$$

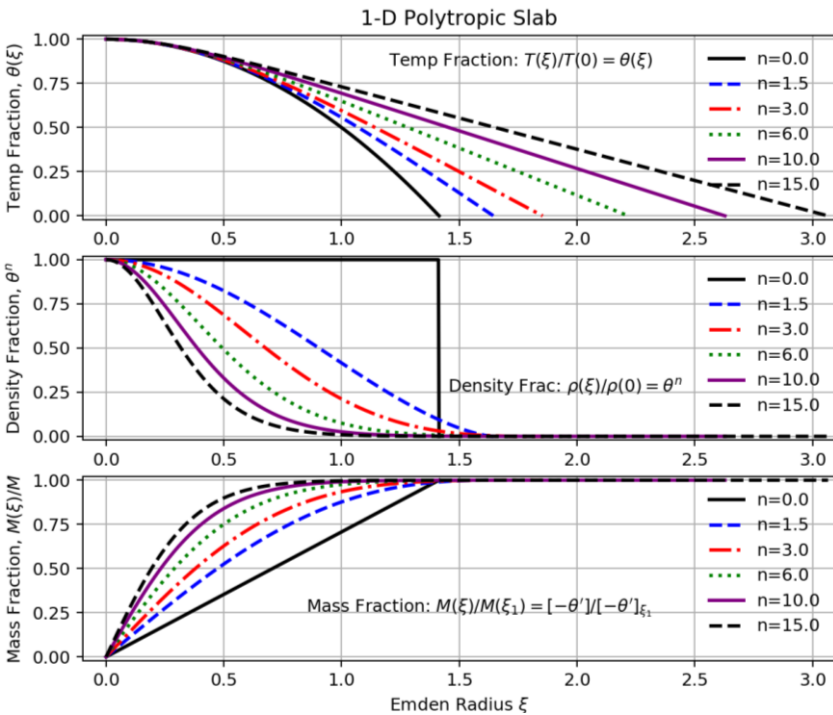
**Table 3.2.1:** 1-D Polytypic Slab Surface ( $\pm\xi_1$ ) Parameter Values

Computing Algorithm: Euler-Richardson (step size: $\Delta\xi = 0.00005$ )				
Index	Radius	Slope	Mass /area param.	Mid Slab Rel.Dens
n	$\pm\xi_1$	$ \theta' _{\xi_1}$	$M(\xi_1) \propto  -\theta' _{\xi_1}$	$\rho_c/\bar{\rho} =  \xi / -\theta' _{\xi_1}$
0.0	1.41421356	-1.41421356	1.41421356	1.00000000
0.5	1.49366840	-1.15470058	1.15470058	1.29355473
1.0	1.57079633	-1.00000000	1.00000000	1.57079632
1.5	1.64534085	-0.89442719	0.89442719	1.83954699
3.0	1.85407468	-0.70710678	0.70710678	2.62205755
5.0	2.10327316	-0.57735027	0.57735027	3.64297597
8.0	2.43165668	-0.47140452	0.47140452	5.15832279
10.0	2.62843161	-0.42640143	0.42640143	6.16421852
15.0	3.06630469	-0.35355339	0.35355339	8.67281936
20.0	3.44944436	-0.30860670	0.30860670	11.17747725

As explained in Appendix A and also done in Sec.2.2 the second order ERM is used here also to integrate the 1-D LEEq given by (3.12a). Table 3.2.1 lists the set of four slab surface parameters (similar to those in Table 2.2.2) obtained for ten values of the polytypic index in the range  $0 \leq n \leq 20$ . All of these parameters are in excellent agreement with similar values listed up to all nine decimal digits by Horedt in Table 2.5.2 of [10].

In Figure. 3.2.1, similar radial properties of 1-D polytypic slabs are graphed for  $n = 0, 1.5, 3, 6, 10$  and  $15$ . In the graph, the top  $\theta(\xi)$  is drawn against  $\xi$  which increases with  $n$ . In the middle graph, the density fraction  $\theta^n$  is drawn showing that it increases with  $n$  and moves closer to the central plane of the polytypic slab. The constant density profile of the  $n = 0$  polytrope may also be noted. At the bottom of the graph, the mass fraction (normalised mass distribution) is drawn showing that more mass collects closer to the central plane due to an increase of density with index  $n$  (see Table

3.2.1). Thus, the central condensation also slowly increases with  $n$  while the polytropic slab itself becomes thicker but finite. However, the rate of increase is much lower than that of 2-D polytropic cylinders.



**Figure 3.2.1:** Radial parameter variations of 1-D Polytropic Slabs are shown graphed together in three groups for  $n = 0, 1.5, 3, 6, 10$  and  $12$ . At the top, middle and bottom the temperature fraction  $\theta(\xi)$ , the density fraction  $\theta^n$  and the mass fraction are respectively drawn against  $\xi$  (see text for details).

In this way, the computational work done here with the second order ERA method has enabled solving the 1-D LEEq and computing quite accurate values of the parameters for several indices in fast sequence and also draw an all-in-one graph. Among the many applications of the 1-D polytropic slab model, a recent one is described in [10].

## 4. Discussion and Conclusion

In Sec. 2 of this review paper, the 2-D LEEq describing the structure of polytropic cylinders was derived along with expressions for density fraction, mass fraction and the central (axial) condensation. Similarly, in Sec. 3 the 1-D LEEq describing the structure of polytropic slabs was derived along with other expressions for density fraction, mass fraction and the central (planar) condensation. Both the LEEqs were integrated using the second order Euler-Richardson method for several indices using the Python script listed fully in Appendix B and the computed physical parameters were collected in Table 2.2.2 and Table 3.2.1 respectively. All of these parameters were found to agree very well with Horedt [10]. The 3-D LEEq describing spherical polytropes too was integrated by the same method and the relevant parameters which also agree with all the digits as given in [10], are listed in Table A.1 in Appendix A. Graphs were also drawn in all the three cases (Figures 2.2.1, 3.2.1 and A.1) for ease of comparison and understanding.

It is interesting to notice that while the graphs appear similar for all the three polytrope models, the parameter magnitudes for the same indices reduce rapidly from that of 3-D models to those of 2-D and 1-D models respectively. Also, the respective magnitudes remain finite in the lower dimensional models. There are many astrophysical applications of these models and some of them requiring more work may be found in [5], [9], [10].

A highlight of this paper has been to demonstrate that the second-order Euler-Richardson algorithm combined with a suitable step size is computationally capable of producing highly accurate parameter values for all the polytrope models considered and also is quite fast.

## Acknowledgements

Some python scripting ideas were taken from the class notes of Ehsan Moravveji [11].

## References

- [1] M. N. Anandaram, *On Emden's Polytropes: Gas Globes in Hydrostatic Equilibrium*, Mapana J. Sci., 12, 1, 85-114, 2014.
- [2] M.N. Anandaram, *Evaluation of Scipy.ode Integrators in Solving the Lane-Emden Equation for Polytropes as a BV Problem with a Fitting Method*, Mapana J. Sci., 16, 1, 65-78, 2017.
- [3] J. Ostriker, *The Equilibrium of Polytropic and Isothermal Cylinders*, Ap. J., 140, 1056 -1066, 1964.
- [4] J. Ostriker, *Cylindrical Emden and Associated Functions*, Ap. J. Supp., 11, 167-183, 1965.
- [5] S. Chandrasekhar and E. Fermi, *Problems of Gravitational Stability in the Presence of a Magnetic Field*, Ap. J., 118, 116-141, 1953.
- [6] A. Cromer, *Stable Solutions using the Euler Approximation*, Am. J. Phys., 49, 455, 1981.
- [7] Retrieved from [https://en.wikipedia.org/wiki/Semi-implicit\\_Euler\\_method](https://en.wikipedia.org/wiki/Semi-implicit_Euler_method)
- [8] B.K.Nikolic, Retrieved from [http://www.physics.udel.edu/~bnikolic/teaching/phys660/numerical\\_ode/](http://www.physics.udel.edu/~bnikolic/teaching/phys660/numerical_ode/)
- [9] S. Recchi et al., *Non-isothermal filaments in Equilibrium*, A&A, 558, A27, 2013 (also download from arXiv: 1308.5792v1[astro-ph.GA]).
- [10] G.P. Horedt, *Polytropes: Applications in Astrophysics and Related Fields*, Kluwer Academic Publishers, New York, 2004.
- [11] [https://bitbucket.org/ehsan\\_moravveji/ivs\\_sse/src/master/exercises/exercises07/](https://bitbucket.org/ehsan_moravveji/ivs_sse/src/master/exercises/exercises07/)
- [12] J. Ostriker, *The Equilibrium of Self-Gravitating Rings*, Ap. J., Vol.140, 1067 -1087, 1964.

## Appendix A

Here is a brief description of the stable first order Euler-Cromer Method (ECM aka symplectic Euler and semi-implicit Euler, [6],[7]) and the second order Euler-Richardson Method (ERM, [8]). The scripting was done in Python 3.6 and executed in Anaconda Spyder IDE. The 2-D LEEq which is given by  $\theta'' = -\theta'/\xi - \theta^n$  (2.13), is redefined here as the function,

$$fzdot(\xi, y, z) \equiv z' = -z(\xi)/\xi - (y(\xi))^n \tag{A.1}$$

The variables y and z are defined by  $y \equiv \theta$  and  $z \equiv y' = d\theta/d\xi$ . Integrating  $z' = dz/d\xi$  twice we can write each integrated step in terms of the current and next updated values as,



$$z(\xi + \Delta\xi) = z(\xi) + \Delta\xi \cdot f_{zdot}(\xi, y, z) \quad (\text{A.2})$$

$$y(\xi + \Delta\xi) = y(\xi) + \Delta\xi [z(\xi + \Delta\xi)] \quad (\text{A.3})$$

The two steps (A.2) and (A.3) with  $\Delta\xi$  as the pre-settable step size constitute the core of the first order ECM. Cromer discovered that inserting the just updated value of  $z(\xi + \Delta\xi)$  from (A.2) in (A.3) leads to improved global accuracy and imparts stability to the integration. The difference with the Forward Euler method is that the ECM uses the newer value  $z(\xi + \Delta\xi)$  in (A.3) while the Euler method uses the older value  $z(\xi)$  leading to instability issues and so is not preferred.

The second order ERM similarly adds two mid-step evaluations using the Euler method in (A.4) and (A.5) which are then used to update the two parameters in (A.6) and (A.7) respectively. These four integration steps are given by,

$$z(\xi + 0.5\Delta\xi) = z(\xi) + 0.5\Delta\xi \cdot f_{zdot}(\xi, y, z) \quad (\text{A.4})$$

$$y(\xi + 0.5\Delta\xi) = y(\xi) + 0.5\Delta\xi \cdot z(\xi) \quad (\text{A.5})$$

$$z(\xi + \Delta\xi) = z(\xi) + \Delta\xi \cdot f_{zdot}(\xi + 0.5\Delta\xi, y(\xi + 0.5\Delta\xi), z(\xi + 0.5\Delta\xi)) \quad (\text{A.6})$$

$$y(\xi + \Delta\xi) = y(\xi) + \Delta\xi \cdot z(\xi + 0.5\Delta\xi) \quad (\text{A.7})$$

The set of these four equations should be used in the same order shown and it is found that integration using ERM is as fast as or faster than using ECM as larger step sizes could be employed. In the computing work reported here, the ERM has therefore been employed to integrate the **2-D LEEq** for a few indices and the results obtained are displayed in Table 2.2.1.

In the case of the **1-D LEEq** given by (3.12a), we redefine (A.1) as,

$$f_{zdot}(\xi, y, z) \equiv z' = - (y(\xi))^n \quad (\text{A.8})$$

Notice that (A.8) is used updated in (A.6) as  $f_{zdot}(\xi + 0.5\Delta\xi, y, z) = - (y(\xi + 0.5\Delta\xi))^n$ .

The ERM is applied to solve the 1-D LEEq using the five expressions (A.4) to (A.8). The results obtained are displayed in Table 3.2.1 along with the step size used. Further discussion of the results is continued in Sections 2.2 and 3.2 respectively.

For the sake of completeness, we mention that before applying the ERM to solve the 3-D LEEq of spherical polytropes that (A.1) must be modified by inserting the factor 2 as shown below.

$$fzdot(\xi, y, z) \equiv z' = -2z(\xi)/\xi - (y(\xi))^n \tag{A.9}$$

The integration then proceeds with the four expressions (A.4) to (A.7) as explained above. The results obtained for a few indices of 3-D spherical polytropes are shown in Table A.1 below and are also plotted together in Figure A.1 for enabling comparison with similar properties of polytropic cylinders given in Sec.2 and polytropic slabs gave Sec.3 respectively.

**Table A.1:** 3-D Polytropic Sphere: Radial Parameters

Computing Algorithm: Euler-Richardson with step size $\Delta\xi = 0.00005$				
Index	Emden Radius	Slope at $\xi_1$	Mass	Central Condens.
npol	$\xi_1$	$ \theta' _{\xi_1}$	$M(\xi_1) \propto  \xi^2 \theta' _{\xi_1}$	$\rho_c/\bar{\rho} =  \xi/3\theta' _{\xi_1}$
0.0	2.4494897444	-0.8164965810	4.8989794954	1.0000000000
1.0	3.1415926507	-0.3183098871	3.1415926573	3.2898681208
1.5	3.6537537336	-0.2033012824	2.7140551124	5.9907045204
2.0	4.3528745899	-0.1272486508	2.4110459998	11.4025428728
3.0	6.8968485892	-0.0424297573	2.0182359193	54.1824811855
3.25	8.0189374727	-0.0303219610	1.9498039207	88.1532416562
4.0	14.9715459597	-0.0080180786	1.7972297650	622.4078834608
4.5	31.8364590809	-0.0017145487	1.7377981706	6189.4731792802

The tabulated 2<sup>nd</sup> column values are found to be in very good agreement with the values listed in [2], [1] and all the values are found to completely agree with Horedt (Table 2.5.2 in [10]).

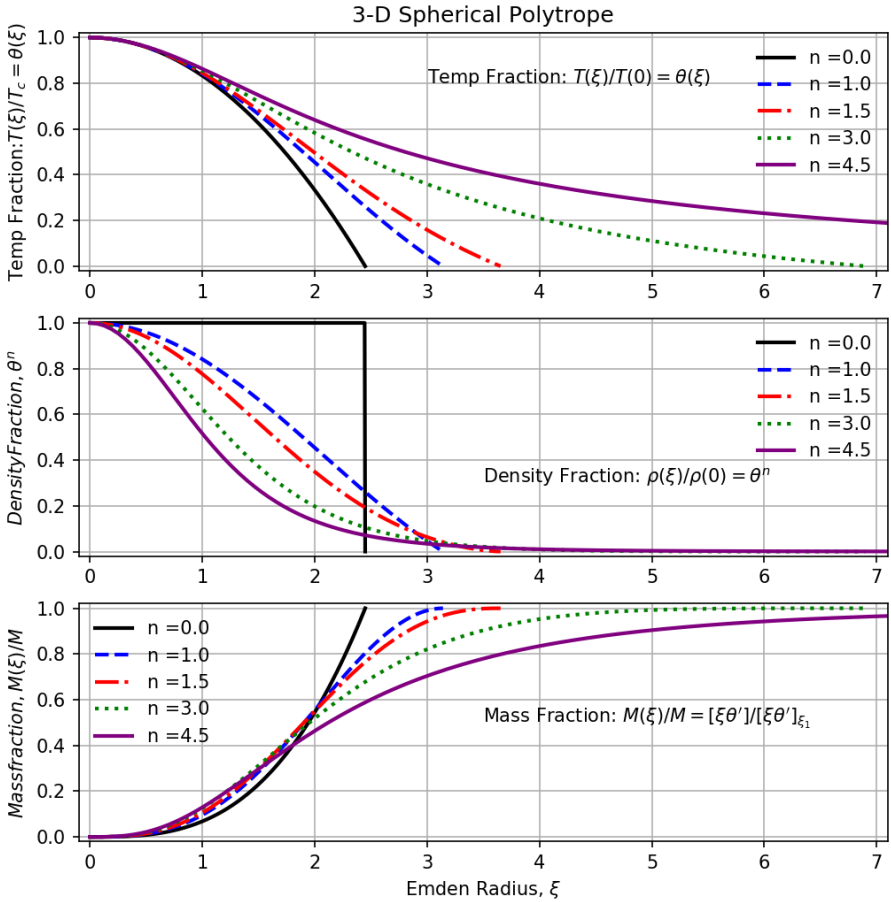


Figure A.1: The temperature, density and mass fractions of 3-D spherical polytropes are graphed together in the top, middle and bottom parts of the figure respectively for a few indices (see [1] for details).

## Appendix B

We can consider huge self-gravitating polytropic rings as an interesting variation in the configuration of the infinitely long 2-D polytropic cylinders treated in Sec.2 above. All the relevant physical parameters have the same radial distribution as discussed in that section with respect to the axis of a non-rotating ring. Ostriker [12] has provided a detailed theory of static as well as solidly rotating rings. Also in [4], he has provided a non-linear DE in terms of the 2-D Emden function  $\theta(\xi)$ , its derivatives (see 2.7, 2.8 and 2.9) and a perturbation function  $f_1(\xi)$  which quantifies the

effects of rotation as well as curvature in distorting the cylindrical structure (see [12] for details). This perturbation function is defined by the non-linear differential equation given by

$$f_1'' + (1/\xi)f_1' + (n\theta^{n-1} - (1/\xi^2))f_1 = -\theta' \tag{B.1}$$

The purpose of this appendix is to numerically solve this equation using mostly the same computer program described in Appendix C. That program is used to supply the run of values of the three parameters, namely, the Emden radius  $\xi$ , the 2-D Emden function  $\theta(\xi)$  and its slope  $\theta'(\xi)$  which are used as **constants** in (B.1) so that it can also be integrated by the same Euler-Richardson Method for a few indices of interest. To do this (B.1) is written as two coupled first-order equations given by,

$$g_1(\xi) = df_1/d\xi, \text{ and, } g_1'(\xi) = (-g_1/\xi - (n\theta^{n-1} - (1/\xi^2))f_1 - \theta') \tag{B.2}$$

The steps in implementing ERM for (B.2) are similar to the set (A.4) to (A.7):

$$fg1dot(\xi, f_1(\xi), g_1(\xi)) \equiv (-g_1(\xi)/\xi - (n\theta^{n-1} - (1/\xi^2))f_1(\xi) - \theta') \tag{B.3}$$

$$g_1(\xi + 0.5\Delta\xi) = g_1(\xi) + 0.5\Delta\xi \cdot fg1dot(\xi, f_1(\xi), g_1(\xi)) \tag{B.4}$$

$$f_1(\xi + 0.5\Delta\xi) = f_1(\xi) + 0.5\Delta\xi \cdot g_1(\xi) \tag{B.5}$$

$$g_1(\xi + \Delta\xi) = g_1(\xi) + \Delta\xi \cdot fg1dot(\xi + 0.5\Delta\xi, f_1(\xi + 0.5\Delta\xi), g_1(\xi + 0.5\Delta\xi)) \tag{B.6}$$

$$f_1(\xi + \Delta\xi) = f_1(\xi) + \Delta\xi \cdot g_1(\xi + 0.5\Delta\xi) \tag{B.7}$$

The two Emden variables  $\theta(\xi)$  and its slope  $\theta'(\xi)$  are held constant in the ERM steps (B.4) to (B.7) and only the Emden radius is changed for updating as shown in the above step sequence. The relevant python function module, `rhs_2D_Rot_ERM()` is provided at line 101 in Appendix C. Along with the index and Emden radius, the Table B.1 lists the three distortion parameters as shown. The values in the last two columns are in excellent agreement with the values reported by Ostriker in various tables in [4].

**Table B.1:** Rotational Distortion Parameters in Polytropic Rings (Ref. [4])

Index	Emden Radius	Slope: $f_1'(\xi_1)$	Distortion Parameter #1	Distortion Parameter #2
n	$\xi_1$	$df_1/d\xi$	$\xi_1 f_1'(\xi_1)$	$f_1(\xi_1)$
1.0	2.4048255572	0.4083503443	0.982011	0.519159
1.5	2.6477767660	0.4455296025	1.179663	0.576370
2.0	2.9213207237	0.4699228976	1.372795	0.654680
3.0	3.5739009830	0.4971514279	1.776770	0.868391
5.0	5.4275745908	0.5142767224	2.791275	1.542781
8.0	10.4029170047	0.5147210132	5.354600	3.466777
10.0	16.2227407555	0.5122874307	8.310706	5.786567
12.0	25.4531322552	0.5100107209	12.981370	9.538241
16.0	63.5140212207	0.5067567412	32.186158	25.416213
20.0	160.5964733254	0.5048037529	81.069702	66.972774

**Appendix C**

The listed 196 line Python script is based on purpose specific function modules, namely, **main()** at line 184, **rhs\_EulerCromer()** at line 71, **rhs\_EulerRichardson()** at line 83, **rhs\_2D\_Rot\_ERm ()** at line 101, **integrate ()** at line 118, **do\_interpolate1 ()** at line 62 and **plot\_EmdenModel ()** at line 5. This script is designed to use either ECM or ERM to solve the 2-D LEEq for polytropic cylinders and to output the parameter values as well as the relevant graph. It can very easily be adapted to solve either the 1-D LEEq for polytropic slabs or the 3-D LEEq for polytropic spheres also by modifying the return statements both at lines 78 and 94 according to the expressions (A.8) or (A.9) in Appendix A. This author can also be contacted by email for any desired assistance.

	# Appendix C
	# PolytropicCylinder_2D_Model.py
	# Python Script written by Mandyam N. Anandaram (run in Python 3.5 or later )
0	# -*- coding: utf-8 -*-
1	import scipy as np
2	from scipy.interpolate import interp1d
3	from matplotlib import pyplot as plt
4	#
5	def plot_EmdenModel( list_dic ) :
6	"""
7	Plot all 2-D cylindrical polytrope solutions

8	"""
9	import itertools # comes with python
10	#
11	fig = plt.figure(figsize=(7,10), dpi=175)
12	top = fig.add_subplot(311)
13	mid = fig.add_subplot(312)
14	bot = fig.add_subplot(313)
15	plt.subplots_adjust( left=0.11, right=0.98, bottom=0.1, top=0.95)
16	colr_list = itertools.cycle( [ 'black', 'blue', 'red', 'green', 'purple' ] )
17	lnstyl_list = itertools.cycle( [ 'solid', 'dashed', 'dashdot', 'dotted' ] )
18	#
19	for i_dic, dic in enumerate( list_dic ) :
20	n = dic["n"]
21	Xi = dic["Xi"]; Xi1 = Xi[-1]
22	Txi = dic["Txi"]
23	d_Txi = dic["d_Txi"]; d_Txi1 = d_Txi[-1]
24	Dxi = Txi**n; Dxi[-1] = 0.0
25	#Pxi = Dxi * Txi; Pxi[-1] = 0.0
26	Mxi = -Xi * d_Txi; Mxi1 = Mxi[-1] # Mxi[Xi1]
27	Mxi /= Mxi1 # normalize mass parameter here
28	print("npol=%2.2f; Xi1 = %3.10f; d_Txi1 = %3.10f" %( n, Xi1, d_Txi1 ) )
29	#lnsty = lnstyl_list.next(); colr = colr_list.next() in py27
30	lnsty = next(lnstyl_list); colr = next(colr_list) # in py36
31	lbl = r'n=%1f%n # + str(dic["n"])
32	top.plot(Xi, Txi, linestyle=lnsty, color=colr, lw=2, label=lbl)
33	mid.plot(Xi, Dxi, linestyle=lnsty, color=colr, lw=2, label=lbl)
34	bot.plot(Xi, Mxi, linestyle=lnsty, color=colr, lw=2, label=lbl)
35	top.set_xlim(-0.2, 18.0) #17.0)
36	#top.set_xticklabellnsty(()) # uncommenting removes xticks in top plot
37	top.set_title(r"2-D Polytropic Cylinder")
38	top.set_ylim(-0.02, 1.02)
39	top.grid();top.text(4.0,0.85,r"Temp Fraction: $T(\xi)/T(0)=\theta(\xi)$ ")
40	top.set_ylabel(r"Temp Fraction, $\theta(\xi)$ ")
41	#top.set_xlabel(r"Emden Radius $\xi$ ")
42	mid.set_xlim(-0.1, 5.0)
43	#mid.set_xlabel(r"Emden Radius $\xi$ ")
44	mid.set_ylim(-0.02, 1.02)
45	mid.set_ylabel(r"Density Fraction, $\theta^n$ ")
46	mid.grid()
47	mid.text(2.12,0.3,r"Density Fraction: $\rho(\xi)/\rho(0)=\theta^n$ ")
48	bot.set_xlim(-0.1, 5.0)
49	bot.set_xlabel(r"Emden Radius $\xi$ ")
50	bot.set_ylim(-0.02, 1.02)
51	bot.set_ylabel(r"Mass Fraction, $M(\xi)/M$ ")
52	bot.grid();
53	bot.text(1.9,0.2,r" MassFraction: $M(\xi)/M=[\xi\theta]/[\xi\theta]_{\xi_1}$ ")

54	leg1 = top.legend(loc="best",frameon=False)
55	leg1 = mid.legend(loc="best",frameon=False)
56	leg1 = bot.legend(loc="best",frameon=False)
57	# save plot in png format
58	plt.savefig( "2D_CylindricalPolytrope.png" )
59	print("New plot saved as '2D_CylindricalPolytrope.png' ")
60	return None
61	#
62	def do_interpolate1 ( XiLst, YxLst, ZxLst ) :
63	#interpolate full range of all huge lists to 501 points
64	Xi_1k = np.linspace( 0.0, XiLst[-1], 501)
65	fXY = interp1d( XiLst, YxLst )
66	Yx_1k = fXY( Xi_1k )
67	fXZ = interp1d( XiLst, ZxLst )
68	Zx_1k = fXZ( Xi_1k )
69	return Xi_1k, Yx_1k, Zx_1k
70	#
71	def rhs_EulerCromer(xi,d_theta,theta,stepxi,npol):
72	"""
73	Use first order Euler-Cromer algorithm for 2-D polytropic cylinders :
74	d_theta[i+1] = d_theta[i]
75	- d_xi*[ (1/xi[i]).(d_theta[i]/d_xi) + theta[i]^n ] <---2D
76	theta[i+1] = theta[i] + d_xi * d_theta[i+1] # ECM used here
77	"""
78	def fzdod ( n, x, y, z ) : return - z/x - y**n # this is for the 2-D case
79	d_theta_next = d_theta + stepxi * fzdod(npol,xi,theta,d_theta)
80	theta_next = theta + stepxi * d_theta_next
81	return d_theta_next, theta_next
82	#
83	def rhs_EulerRichardson(xi,d_theta,theta,stepxi,npol):
84	"""
85	Use 2nd order Euler-Richardson algorithm for 2-D polytropic cylinders :
86	d_theta[i+1/2] = d_theta[i] - 0.5*d_xi*[d_theta[i]/xi[i] + theta[i]**npol]
87	theta[i+1/2] = theta[i] + 0.5*d_xi * d_theta[i] #
88	d_theta[i+1] = d_theta[i] -
89	d_xi*[d_theta[i+1/2]/xi[i+1/2] + theta[i+1/2]**npol ] <-2-D
90	theta[i+1] = theta[i] + d_xi * d_theta[i+1/2] #
91	return d_theta[i+1], theta[i+1]
92	"""
93	stepxi_mid = 0.5 * stepxi;    xi_mid = xi + stepxi_mid
94	def fzdod ( n, x, y, z ) : return - z/x - y**n # this is for the 2-D case
95	d_theta_next_mid = d_theta + stepxi_mid*fzdod(npol,xi,theta,d_theta)
96	theta_next_mid = theta + stepxi_mid * d_theta
97	d_theta_next= d_theta +
98	stepxi*fzdod(npol,xi_mid,theta_next_mid,d_theta_next_mid)
98	theta_next = theta + stepxi*d_theta_next_mid

99	return d_theta_next, theta_next
100	#
101	def rhs_2D_Rot_ERm ( X, Y, Z, npo ) :
	# use ERM for Rotational Distortion params of 2-D polytropic Rings
	nrws = len(X); dxi = X[2]-X[1]; dxi2 = dxi/2.0 #const step size
	xg1Lst = [xg1]; f1Lst = [f1]; df1Lst = [0.0]; npo1 = npo - 1.0
	for ipt in range ( nrws):
	xi = X[ipt]; ytx = Y[ipt]; zdtx = Z[ipt]; xi15 = xi + dxi2
	if ipt == 0: xi = 0.00000001
	g1_nxt2 = g1 + dxi2*(-g1/xi- f1*(npo*ytx**npo1 - 1.0/xi/xi) - zdtx)
	f1_nxt2 = f1 + dxi2 * g1
110	g1_nxt = g1 + dxi*(-g1_nxt2/xi15- f1_nxt2*(npo*ytx**npo1 - 1.0/xi15**2) - zdtx)
	f1_nxt = f1 + dxi * g1_nxt2
	g1 = g1_nxt; f1 = f1_nxt
	df1Lst.append(g1); xg1Lst.append(xi * g1); f1Lst.append(f1)
	f1Lst = np.array(f1Lst); df1Lst = np.array(df1Lst);
	xg1Lst = np.array(xg1Lst);
116	return df1Lst, xg1Lst, f1Lst
	#
118	def integrate ( n = 3.0 ) :
	"""
120	integrate the Lane Emden equations, and
	return the solution as a dictionary
	"""
	if n > 20.01: raise SystemExit("Achtung: n =%.3f > 20.0 limit!"%n)
	# Define empty list containers for various parameters
	Xi_list = [] # container for emden radii xi
	Txi_list = []; # container for the L-E function, theta(xi)
	d_Txi_list = [] # container for slope dtheta/dxi
	# Initial Conditions at the center of polytrope
	xi, Txi, d_Txi = 0.0, 1.0, 0.0 #Dxi = 1.0; Pxi = 1.0; Mxi = 0.0
130	# Stepsize
	xi_step = 0.00002
	# append (store) the central values to the respective lists
	ic = 0
	Xi_list.append(xi)
	Txi_list.append(Txi)
136	d_Txi_list.append(d_Txi)
	#
	ic = 1 # compute theta and d_theta from truncated power series
	xi += xi_step
140	Xi_list.append(xi)
	Txi = 1.0 - xi**2/4.0 + xi**4 * n/64.0 - xi**6 * n*(3*n-2)/2304.0
	Txi_list.append(Txi)
	d_Txi = -xi/2.0 + xi**3 * n/16.0 - xi**5 * n*(3*n-2)/384.0
144	d_Txi_list.append(d_Txi) # slope of theta = d_theta/d_xi



	<code>#d_dtheta = -1.0/2.0 + xi**2 * 3*n/16.0 - xi**4 * 5*n*(3*n-2)/384.0</code>
146	<code>#print '%4d % .11f % .11f % .11f'(i,xi,theta,d_theta)</code>
	<code>#</code>
148	<code>Txi_next = Txi</code>
	<code>#</code>
150	<code>while Txi_next &gt;= 0.000001 :</code>
	<code>    #d_Txi_next, Txi_next = rhs_EulerCromer(xi,d_Txi,Txi,xi_step,n)</code>
	<code>    d_Txi_next, Txi_next = rhs_EulerRichardson(xi,d_Txi,Txi,xi_step,n)</code>
	<code>    if (Txi_next &gt; Txi): raise ValueError('theta increasing:', ic, xi, Txi,</code>
	<code>    Txi_next)</code>
	<code>    ic += 1</code>
155	<code>    xi += xi_step</code>
	<code>    Txi = Txi_next; #--&gt;temperature, T/Tc = theta</code>
	<code>    d_Txi = d_Txi_next</code>
	<code>    Xi_list.append(xi)</code>
	<code>    Txi_list.append(Txi)</code>
160	<code>    d_Txi_list.append(d_Txi)</code>
	<code>#</code>
162	<code>print ( "Solved for n = {0} after {1} steps of {2}".format( n, i, xi_step ) )</code>
	<code>Xi_list = np.array(Xi_list)</code>
	<code>Txi_list = np.array(Txi_list)</code>
	<code>d_Txi_list = np.array(d_Txi_list)</code>
	<code>#print(n,Xi_list[-2],Xi_list[-1]);</code>
	<code>#print(n,Txi_list[-2],Txi_list[-1]); print(n,d_Txi_list[-2],d_Txi_list[-1])</code>
	<code># Set theta=0 and Compute by linear extrapolation xi1,&amp; d_Theta at Xi1.</code>
	<code>Xi_list[-1] = Xi_list[-2] - Txi_list[-2]/d_Txi_list[-2]</code>
170	<code>Xi1 = Xi_list[-1] # Emden radius</code>
	<code>d_Txi_list[-1] = d_Txi_list[-2] + (Xi_list[-1]-Xi_list[-2])*(-</code>
	<code>    d_Txi_list[-2] / Xi_list[-2] - Txi_list[-2]**n) #2-D</code>
	<code>Txi_list[-1] = 0.0 # theta = 0 at edge/surface of polytrope, xi1</code>
	<code>d_Txi1 = d_Txi_list[-1]</code>
	<code>print ( "n=%2.2f: xi1 = %.10f; Mxi1 = %.10f; ccn = %.10f" (n,</code>
	<code>    Xi1, -Xi1*d_Txi1, -Xi1/2.0/d_Txi1) )</code>
177	<code>Xi_1k, Txi_1k, d_Txi_1k = do_intrplt1(Xi_list, Txi_list, d_Txi_list)</code>
	<code>#uncomment below only if want to compute rotational distortion params</code>
179	<code>#df1Lst, xg1Lst, f1Lst = rhs_2D_Rot_ERm(Xi_1k, Txi_1k, d_Txi_1k, n)</code>
180	<code>#print("df1Xi1 = ",df1Lst[-1],"; Xi1G1 = ",xg1Lst[-1], "; F1xi1 = ",f1Lst[-1])</code>
181	<code>dic = {"n":n, "Xi":Xi_1k, "Txi":Txi_1k, "d_Txi":d_Txi_1k }</code>
182	<code>return dic</code>
	<code>#</code>
184	<code>def main () :</code>
	<code>    """</code>
	<code>    The main caller which kick starts the computational engine above.</code>
	<code>    """</code>
	<code>    n_arr = [ 0.0, 1.5, 3.0, 5.0, 10.0, 15.0 ] # used for 1-D and 2-D polytropes</code>
	<code>    list_dic = [] # empty list to hold all arrays for plotting etc.</code>

190	for npol in n_arr : list_dic.append(integrate(npol)) # list is now filled
191	plot_EmdenModel ( list_dic ) # draw graphs using arrays from the list
192	plt.show() # display the graph
193	return None
194	#
195	if __name__ == '__main__':
196	status = main() # this starts the main caller at line 164 above till 172