

HERCULES version 1.0 tutorial

Simon J. Lock

Last updated: February 14, 2019

1	Introduction	2
1.1	Overview of HERCULES	2
2	Running HERCULES	4
3	Understanding the output of HERCULES	5
3.1	Command line output	5
3.2	Structure files	7
4	Plotting	11
5	Rotating planets	12
5.1	Constant rotation	12
5.2	Angular momentum conservation	13
6	Mass, composition and thermal state	13
7	Initial conditions	14
8	Precision	14
8.1	Iteration tolerances	14
8.2	Number of layers	15
8.3	Truncating spherical harmonic expansion	15
9	Conclusion	15

1 Introduction

This tutorial provides an introduction to HERCULES v1.0 and its basic functionality. It is not intended to provide rigorous instruction on using HERCULES and it is recommended that users intending to use HERCULES at more than an introductory level should also read the users manual.

To complete this tutorial you will need a compiled version of HERCULES and the files that accompany this guide. Consult the users guide for notes on compiling HERCULES. To use the analysis scripts provided with this tutorial you will need to have python installed. To test you have all the libraries required for this tutorial run the script `test_python_libraries.py` that accompanies this tutorial.

1.1 Overview of HERCULES

HERCULES - short for Highly Eccentric Rotating Concentric U (potential) Layers Equilibrium Structure - is a code designed to find the equilibrium structure of fluid planets including bodies that are rotating rapidly. The theoretical framework for the code is based on work originally done for studying Jupiter [Hubbard, 2012, 2013] but extended to overcome the problems of numerical convergence in modelling highly oblate bodies [Hubbard et al., 2014; Kong et al., 2013].

In HERCULES, a planet is modelled as a series of concentric constant density layers, made by superimposing constant density spheroids (see box below for notes on terminology and Figure 1). The gravitational potential due to each of the spheroids can be calculated and summed to give the total potential at a point in the structure as a expansion in spherical harmonics. Using this useful fact, the shape of each of the spheroids is iterated until the surface of each of the spheroids is an equipotential surface, i.e. to give the equilibrium structure of the planet. The code can also iterate to conserve the mass of a number of material layers and the total angular momentum of the structure.

The surface of each of the spheroids, and hence layers, is described by a number of points, linearly spaced in $\mu = \cos\theta$, where θ is the angle from the rotation axis. The material layers (e.g. core, lower mantle, upper mantle etc.) are treated as real materials with equations of state (EOSs) that can be used to give self-consistent densities for each of the concentric layers. Spheroids, layers and materials are numbered from the outside in starting from 0. A layer is considered as being in a material layer if it's outside surface is within or at the edge of that material layer.

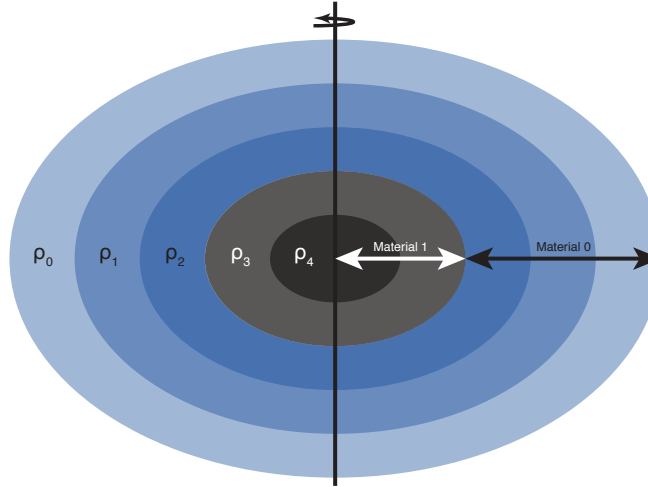


Figure 1: Schematic of an axisymmetric, rotationally-flattened planetary structure modelled by the HERCULES code showing a plane through the rotation axis. The modelled body consists of a number of overlapping spheroids, each with constant density, $\delta\rho_i$. The sum of the density of all the individual spheroids overlapping a given layer gives the total density of that layer, ρ_i . The body is also divided into a number of material layers (shown separately in blues and greys) with the relationship between pressure and total density in that material layer determined by the material equation of state.

Note on nomenclature

In this users guide we use a number of different terms to describe the concentric layers, material layers etc.. Here we attempt to clarify the meaning of each of these.

spheroid or concentric spheroid:

This refers to the concentric spheroids that make up a planet in HERCULES. We use this term to refer to the whole spheroid, which extends from the origin out to the surface of the spheroid. Each spheroid has a density $\delta\rho_i$.

layer or concentric layer:

A concentric layer is the volume that is bounded by the surfaces of two adjacent spheroids. The density of a layer, ρ_i , is given by the sum of all the spheroids that intersect the layer. The properties of the layer are the 'real' properties of the planet described by HERCULES.

material layer:

A material layer is a compositional or thermal layer in a planet e.g. core, lower mantle, upper mantle etc. A material layer is the region of the planet that is considered to be of a given material. The 'real' density of the layers in this region are calculated using the EOS of that material. A layer is considered to be part of a given material layer if its outer surface lies within, or at the top of, the material layer.

2 Running HERCULES

To demonstrate how to run HERCULES we will look at a simple example, that of a non-rotating Earth. This will give us a chance to explore some of the basic options for HERCULES. In order to run, HERCULES requires an input file called `HERCULES_input.txt` and an output directory `Output` in the run directory. HERCULES also needs EOS files but the path can be provided for these in the impact directory.

Open the example input file `HERCULES_input_tutorial1.txt`. You will notice that there are a variety of options in the input file. HERCULES reads in the input file line by line. It is imperative that the format of the file is maintained for correct reading with the option name followed by white space, equals sign, white space and then the variable value. For options that require multiple impacts the options should be given as an array with braces with each element separated by commas. In cases where the inputs are strings leave white space between the strings and commas.

We will now look at some of the options and return to look at the other options later. Leave all the options the same for now.

run_name: The name of the run, or alternatively the name of the body that is being solved for. It is used to name the output files and as an identifier within the output files.

flag_Mconc: The mass conservation flag. There are three options for conserving, or not, the mass of a body. For this tutorial we will only use the option 2, conserving mass by scaling the radius of all of the spheroids. For more detail see the users manual.

flag_Lconc: The angular momentum conservation flag. There are two possible options: to have a fixed constant rotation rate or conserve angular momentum by adjusting the rotation rate of the body. For now we will use the first of these options.

Nmaterial: The number of material layers in a body. For now we will just consider two possible layers, mantle and core.

material_files: The EOS files for each of the material layers. For this tutorial we will use the precomputed EOS files provided with this tutorial. You will need to make sure that the paths given as input here is correct for where you have put this tutorial relative to your run directory.

omega_rot: The constant angular velocity for the body used for our selected angular momentum conservation flag.

Try running HERCULES using the example input file `HERCULES_input_tutorial1.txt` without changing any of the input parameters. You can do this by simply running

```
1 ./HERCULESv1.0 > output.txt
```

in the command line. Remember to rename the input file to `HERCULES_input.txt` before running. If the run is a success then the final line of the output line in the `output.txt` should read `run time (s)` followed by a run time. On my 2012 MacBook Pro this example run takes ~ 10 s. If the run takes significantly longer than this then you might need to check that the paths to the EOS files are correct.

Congratulations, you have made your first planet. In the Output directory you should see two new files. These are HERCULES output files. In the next section we will look at this output.

3 Understanding the output of HERCULES

The output for HERCULES comes in two parts, the command line output and binary structure files. We will now look at each of these and explain what they mean.

3.1 Command line output

The command line output is designed to give details about the progress of the code and, in particular, the iterations that are required to find the equilibrium structure. We will break this up to consider each part.

First the start of the file gives information about reading the input file and initialising the structure.

```
1 HERCULES is running
2 Initialising
3   Reading input file
4   Run name: tutorial1.L0_N50_Nm400_k6-f020-p10.l1.0-1.5
5   Input file read successful
6   Initialising with ellipsoids
7     Using evenly spaced mu points
8     Min density 61.962(SI)
9     Max density 24786(SI)
10    Constant rotation with omega = 0
11    Initialisation complete
```

Most of the lines are self explanatory but I will highlight a few.

Line 4: The name of the file is based on the run name and the properties of the run. For an explanation of this see the users manual.

Line 6: For this simple example we have used a very simple initialisation routine that initialises the spheroids as each having a constant density. We will look at more options later in 7. The information in the following indented block gives the resulting maximum and minimum density in the structure and tells us that we are using a constant given rotation rate.

The second section of the file deals with the various iterations used to find the equilibrium structure.

1	Iterating										
2	Main Iteration #		convergence		a	aspect	ellipticity		l	omega_	
	rot	Mass error	AM error								
3	Mass/AM Iteration #		convergence		p_core	a	omega_rot	edge		corot	
	AM error										
4	1	1	4.40824e+11	7.84047e+06	0	0	nan				
5	2	0.130498	3.89938e+11	6.96949e+06	0	0	nan				
6	3	0.0858397	4.26553e+11	6.64456e+06	0	0	nan				
7	4	0.00403067	4.2828e+11	6.53605e+06	0	0	nan				
8	5	0.0199494	4.19903e+11	6.49422e+06	0	0	nan				
9	6	0.0168073	4.12962e+11	6.47517e+06	0	0	nan				
10	7	0.0103843	4.08718e+11	6.46527e+06	0	0	nan				
11	8	0.00565742	4.06419e+11	6.45968e+06	0	0	nan				
12	9	0.00301187	4.05198e+11	6.45705e+06	0	0	nan				
13	10	0.00190274	4.04429e+11	6.45588e+06	0	0	nan				
14	11	0.00102383	4.04015e+11	6.45518e+06	0	0	nan				
15	12	0.000511465	4.03808e+11	6.45475e+06	0	0	nan				
16	13	0.000259489	4.03704e+11	6.4545e+06	0	0	nan				
17	14	0.000135397	4.03649e+11	6.45436e+06	0	0	nan				
18	15	7.25618e-05	4.0362e+11	6.45428e+06	0	0	nan				
19	16	3.95117e-05	4.03604e+11	6.45424e+06	0	0	nan				
20	17	2.16648e-05	4.03595e+11	6.45422e+06	0	0	nan				
21	18	1.1907e-05	4.0359e+11	6.4542e+06	0	0	nan				
22	19	6.54797e-06	4.03588e+11	6.4542e+06	0	0	nan				
23	20	3.60123e-06	4.03586e+11	6.45419e+06	0	0	nan				
24	21	1.98056e-06	4.03585e+11	6.45419e+06	0	0	nan				
25	22	1.08922e-06	4.03585e+11	6.45419e+06	0	0	nan				
26	23	5.99022e-07	4.03585e+11	6.45419e+06	0	0	nan				
27	1	0.310561	6.45419e+06	1 nan	nan	0	4.33676e-07	nan			

HERCULES has a main iteration loop that contains two more separate iterations. The first of the sub-iterations is to find the shape of each of the spheroids but keeps the equatorial radius of each of the spheroids the same. The second iteration changes the equatorial radii and rotation rate of the spheroids to conserve mass and angular momentum. The first of these loops applies to each of the individual points on each of the layers and so is not recorded in the command line output. However, key properties of the structure are given at each iterative step for the conservation and main loops. Line 2 gives the header

information for the main loop and Line 27 give an example output for a main iteration. All values are in base SI. The values given are as follows:

Iteration #: The number of the iteration.

convergence: The value of the convergence criteria at that iteration.

a: The equatorial radius of the outermost spheroid.

aspect: The ratio of the polar radius to the equatorial radius.

ellipticity: A measure of the flattening of the body. See definition in user manual.

l: A measure of the flattening of the body. See definition in user manual.

omega_rot: The angular velocity of the body.

Mass error: The fractional difference between the current mass of the body and the target mass.

AM error: As above but for angular momentum.

Line 3 gives the header information for the conservation loop and lines 4-26 give the results of a series of iterations for the conservation routine. The values given are some of those given for the main loop with the addition of the following

p_core: Pressure at the centre of the planet.

edge corot: The layer at the edge of a corotating region. We will not be concerned with this for this tutorial.

Note that in the example we are using the angular momentum error appears as NaN as the body is non-rotating and so the fractional error in angular momentum is not well defined.

3.2 Structure files

The main output of the code are binary structure files. These contain all the information about a structure and the run that generated it. Here we will describe the structure of the output files and then look at extracting a few of these.

The first part of a structure file holds the information about the run that generated the structure. This is contained in a parameter class object which has the following variables

Table 1: Structure of the parameter class.

Name	Symbol	Description
<hr/> Variables <hr/>		
<code>run_name</code>		Name of the code run (up to 80 characters)
<code>nint_max</code>	n_{int}	Maximum number of iterations for main iteration and mass-AM iteration
<code>toll</code>	χ_{toll}	Desired fractional change in iteration convergence criteria desired for main iteration and mass-AM iteration
<code>xi_int_max</code>	n_{int}^{ξ}	Maximum number of iterations for shape iteration
<code>xi_toll</code>	χ_{toll}^{ξ}	Desired fractional change in iteration convergence criteria desired for shape iteration
<code>dxi</code>	$\delta\xi$	ξ step used to calculate the $dU/d\xi$ used in shape iteration
<code>flag_start</code>	$\mathcal{F}_{\text{start}}$	Flag for how to initiate the structure in the code. We will look at this in Section 7
<code>start_file</code>		File to use in some initialisation routines. Again we will discuss this in Section 7
<code>flag_Mconc</code>	$\mathcal{F}_{\text{Mconc}}$	Flag for mass conservation option. For this tutorial, and almost all cases of running HERCULES, we will only use flag 2.
<code>flag_Lconc</code>	$\mathcal{F}_{\text{Lconc}}$	Flag for AM conservation option. We will explore this options in Section 5
<code>flag_iter_print</code>	$\mathcal{F}_{\text{print}}$	Flag for whether to print an output file on every iteration. This will be useful later.
<hr/> Vectors <hr/>		
<code>omega_param</code>	λ_i	Parameters for angular velocity profile used in some AM conservation options. This is generally not needed and should just be ignored for the purposes of this tutorial.

These are basically the input file parameters but retained for permanent storage in association with the structure generated. We haven't looked at all these yet but we will introduce them as we go through the rest of the tutorial.

The rest of the structure file contains information about the structure itself. This is stored as a planet class which describes the whole structure and contains within it arrays of layer classes that describe the properties of the spheroids and layers. The parameters in the planet class are as follows.

Table 2: Structure of the parameter class.

Name	Symbol	Description
<hr/>		
Variables		
Nmu	N_μ	Number of μ point used to describe the surfaces of the spheroids
Nlayer	N_{lay}	Number of concentric spheroids used to describe structure
Nmaterial	N_{mat}	Number of different material layers there are in the structure
kmax	k_{max}	Maximum value of k used in the spherical harmonic expansion, see Section 8. This is equivalent to a maximum spherical harmonic degree of $2k_{\text{max}}$
Mtot	M	Total mass of the body. This is the actual mass of the body, not necessarily equal to the target mass.
Ltot	L	Total angular momentum of the body. Again the true angular momentum, not the target angular momentum.
omega_rot	ω_{rot}	Rotation rate of corotating section of planet
pmin	p_{min}	Imposed pressure at the outside of the outermost layer of the structure. This is required for HERCULES but can be set to zero.
amax	a_0	The equatorial radius of the outermost layer
aspect	\mathcal{A}_0	Aspect ratio, ratio of the equatorial to polar radii, of the outermost layer
ref_rho	ρ_{ref}	Reference density for some mass conservation routines. We will not use this here.
Mtot_tar	$\sum M_{\text{tar}}^i$	Target mass of the whole body for mass conservation routines
Ltot_tar	$\sum L_{\text{tar}}^i$	Target AM of the whole body for AM conservation routines
Ucore	U_{core}	Total potential at the centre of the body
pcore	p_{core}	Pressure at the centre of the body
<hr/>		
Vectors		
real_rho	ρ_i	The 'real' densities of each of the layers
press	p_i	Pressure at the top of each of the layers
dldr	dp/dr	Equatorial pressure gradient
mu	μ_i	The array of μ used to describe the surfaces of each spheroid
Ulayers	$U(a_i, 0)$	Total potential of each layer
Mout	M_{out}^i	Mass of material that is further from the rotation axis than the equatorial radius of each layer. We will not use this here.

continued on next page

Table 2 – continued from previous page

Lout	L_{out}^i	As above but for angular momentum
Js	J_{2k}	The gravitational moments for the whole structure
material_lay	$N_{\text{lay}}^{\text{mat}}$	Number of layers in each material layer
layers		Vector of concentric_layer classes, see below, containing the details of each of the layers and spheroids in the planet
materials		Array of EOS classes containing the EOS for each of the materials. We will not deal with this in this tutorial
flag_material		Flag for each layer as to what material it belongs. Starting from 0 for the outside material
M_materials	M_{tar}^i	The target mass for each of the material layers so they can be conserved separately

In the planet structure the layers and spheroids are described using a layer class. This class contains all the information about both the spheroids and the layer that that spheroid describes the outer surface of. The properties in the class are as follows:

Table 3: Structure of the concentric_layer class.

Name	Symbol	Description
Variables		
Nmu	N_{μ}	Number of μ point used to describe surfaces. Same as in planet class.
kmax	k_{max}	Maximum value of k considered. This is equivalent to a maximum spherical harmonic degree of $2k_{\text{max}}$. Same as in planet class.
omega	ω_i	Angular velocity of layer. This could vary depending on assumption about rotational profile but for the purposes of this tutorial is the same for each layer.
rho	$\delta\rho_i$	Density of the concentric spheroid
M	M_i	Mass of the concentric spheroid
V	V_i	Volume of the concentric spheroid
a	a_i	Equatorial radius of the spheroid and outer edge of layer
b	b_i	Polar radius of the spheroid
I	I_i	Moment of inertia of the spheroid

continued on next page

Table 3 – continued from previous page

Vectors		
mu	μ_j	μ for each of the points on the layer. Same as in planet class.
xi	$\xi_i(\mu_j)$	The radius of the surface of the spheroid at each of the μ points, normalised to the equatorial radius of the surface. We will use this to plot the structure in the following Section.
Js	J_{2k}	The scaled gravitational moments for the layer. See the user manual.

Now that we have introduced the structure of the output file, let's extract some information. Accompanying this tutorial, and the distribution of HERCULES, is the python script `HERCULES_structures.py` that contains a description of the above structures and functions to read them in. Also accompanying this tutorial is a script `read_HERCULES_output.py` that contains an example of reading a structure file and extracting some information. Alter this script to read in the output file you generated in Section 2 and answer the following questions. Note, you will want the file `tutorial11_L0_N50_Nm400_k6_f020_p10_l1_0_1.5_final` as this the output at the end of HERCULES. The file ending `_0` is the initial conditions.

1. What is the number of material in the structure? Is it the same as in the input file? (IT SHOULD BE!!!)
2. What is the angular momentum of the planet?
3. What is the pressure at the core of the planet?
4. What is the total moment of inertia of the body? Is this reasonable?
5. What is the mass of the outermost spheroid? Is this the same as the mass of the planet? Does that surprise you?
6. What is the equatorial radius of the core?
7. What is the pressure at the core-mantle boundary? (If you struggle with this one there is a function in `HERCULES_structures.py` that will show you how to do it)

You might notice that this planet looks a lot like Earth.

4 Plotting

Now we have learnt how to extract individual values from a structure file we will now look at plotting various profiles. Accompanying this tutorial is the python script `plot_HERCULES_structure.py`.

This script will plot the shape of the planet and the midplane pressure profile. Run the script for the output from Section 2. Now you are going to modify this script to plot different properties of the planet.

1. In panel A of the plotting script the surfaces of all the layers are the same color. Modify the script to color the core and mantle layers differently.
2. Add a panel with the density profile through the planet. Be careful as you might need to change the relative height of the plots to keep the x-axis the same in each plot.
3. Plot the shape and profiles of the initial structure that HERCULES began the iteration with, the output file ending `_0`, to compare the two. You might want to only plot a select number of spheroids. What do you notice?

5 Rotating planets

We will now turn to using HERCULES to find the structure of rotating planets. There are two ways to do this in HERCULES; fixed rotation rate or conservation of angular momentum. We will look at both of these.

5.1 Constant rotation

Run HERCULES using the example input file `HERCULES_input_tutorial2.txt` without changing any of the input parameters. This will take quite a bit longer to run than for the non-rotating planet we have considered previously. The planet we are solving for is the same as we have looked at previously but with an angular velocity of $2 \times 10^{-4} \text{ rad s}^{-1}$. We have enforced this by using the *flag_Lconc* for constant ω and changing the parameter *omega_rot* in the input file to the desired angular velocity. We have also changed some other parameters to make sure the structure can be found easily. Using the scripts from the previous section consider the following.

1. What is the flattening of the body? What happens if you increase the rotation rate to $3 \times 10^{-4} \text{ rad s}^{-1}$?
2. What is the angular momentum of the body? How does it compare to the present day Earth (you can approximate the angular momentum of the Earth based on your answers in Section 3.1)
3. Plot the final structure as in the previous Section. Compare it to the non-rotating planet. What do you notice? How is this affected by changing the enforced angular velocity?

5.2 Angular momentum conservation

Run HERCULES using the example input file `HERCULES_input_tutorial3.txt` without changing any of the input parameters. This will take quite a bit longer to run than for the non-rotating planet we have considered previously. The planet we are solving for is the same as we have looked at previously but now we are forcing it to have an angular momentum of $5 \times 10^{34} \text{ kg m}^2 \text{ s}^{-1}$. To do this we have used the *flag_Lconc* for conservation of angular momentum and changed the parameter *Ltot* in the input file to the desired angular momentum. Using the scripts from the previous section consider the following.

1. What is the angular velocity of the body?
2. What is the flattening of the body? How does this compare to the bodies with an enforced angular velocity?
3. Plot the final structure and compare it to the non-rotating planet. Are there any differences?

6 Mass, composition and thermal state

You will now learn how to change the mass, composition and thermal state of a planet in HERCULES. The mass of the planet in HERCULES is set by the sum of the masses of the material layers given in the input file as the *Mass* array. In the previous examples we have used an Earth-like planet with $\sim 1/3$ core by mass. Run HERCULES using the example input file `HERCULES_input_tutorial1.txt` but try changing the mass of each of these layers. What is the effect on the size of the body? Note, you will want to change the name of the run so not to overwrite your previous output. If you push the code too far it might fail to find an equilibrium structure. This is because the starting structure is too far from the final structure. We will deal with these issues in Section 7.

Changing the thermal state is also quite easy. For the version of HERCULES used in this tutorial the thermal state of a material layer is set by the EOS used. The EOS for the materials in the planet are set by the *material_files* option in the input file. So far in this tutorial we have used EOS that are approximately that of the present-day Earth. With this tutorial we have provided another pre-computed EOS for a hot, silicate mantle `HERCULES_EOS_forsterite_S6.50v_log.txt`. This thermal structure is for a supercritical mantle grading to pure vapour at lower pressures. Do the following.

1. Try swapping out the mantle EOS for this hotter EOS in input file `HERCULES_input_tutorial1.txt`. What does this do to your planet? Plot the cold and the hot planet together and compare the two.
2. Use input file `HERCULES_input_tutorial2.txt` and consider a hot rotating planet.

Again, plot the cold and hot planets together and compare them. How does the flattening of the planets compare?

7 Initial conditions

So far in this tutorial we have been initialising our planets using the spheroids *start_flag*. This is a very simple routines that initiates the spheroids as simple ellipsoids equally spaces in radius with a material layer. The oblateness of the spheroids is given by the *aspect* parameter in the input file and the outer radius of the material layers to start with is given by the *Nmaterial_amax* array. The density of each of the spheroids within a material layers is the same and chosen to give the correct mass for each material. This routine is simple but can lead to very unrealistic starting conditions. We will now look at altering the starting conditions using this routine and consider starting from an input file.

1. Take `HERCULES_input_tutorial2.txt` and change the initialised aspect ratio and initial radius of the materials. How does this affect how rapidly HERCULES converges? Can you find starting conditions that make the code fail?
2. Compare the final structures found using different starting conditions. How much do they vary?
3. Take `HERCULES_input_tutorial3.txt` and change the *start_flag* to 1 and the target angular momentum to be 2×10^{34} . This means you will start the code using a previously calculated structure file. You will need to specify that files location as *start_file* in the input file. I would suggest you used the final output of the previous task. How does this affect the time taken to converge compared to starting from a given set of spheroids with an aspect ratio of 1? Compare the initial conditions for the two runs by plotting them. Do you see why convergence might be easier?

8 Precision

HERCULES has a number of options that affect the precision of the structure calculated by the code. These include changing the number of spheroids used to construct the planet, the iteration tolerance and the number of terms used in the expressions used to calculate the potential. We will now look at each of these.

8.1 Iteration tolerances

There are two iteration tolerances that can be changed in the input file. *toll* controls the main and conservation iterations and *xi_toll* controls the shape iteration. Generally I recommend keeping *xi_toll* smaller than *toll* to guarantee convergence. Take one of the

example input files and try changing the iteration tolerances. If you require too much precision the code might not converge to that precision with the number of iterations allowed. To increase the number of iterations for the main and conservation loops change *nint_max* in the input file. You shouldn't need to change the number of shape iterations *xi_nint_max*. What is the effect on computation time? What happens if you set the tolerances too low?

8.2 Number of layers

The precision of the structure is also dependent on the number of spheroids used to describe the body. In order to change the number of spheroids you need to change two inputs. *Nlayers* must be set to be the total number of spheroids (or layers) and *Nmaterial_layer* must be given to designate how the layers are divided between the material layers. The sum of the latter must equal the former or the code will not run. Try running with different numbers of layers. What effect does this have on the structure calculated? How does the run time change with the number of layers? Plot the run time vs the number of layers.

8.3 Truncating spherical harmonic expansion

The equations used in HERCULES to calculate the potential rely on an expansion in spherical harmonics (see the user manual). To make the problem numerically tractable the expansion is truncated at a given spherical harmonic degree $2k_{max}$. k_{max} can be set by setting *kmax* in the input file. However, you should be wary playing with this particular parameter. In order for the code to be able to resolve higher order you will need to increase the number of points used to describe the surface of the spheroids, *Nmu* in the input file.

1. Try running HERCULES for a planet rotating relatively slowly (angular velocity $\sim 2 \times 10^{-4} \text{ rad s}^{-1}$) varying *kmax* but leaving *Nmu* the same. What do you notice changes about the structure? Try plotting the effect of *kmax* on the shape of the structure. What happens to the run time?
2. Repeat the exercise but for a rapidly rotating planet (angular velocity $\sim 6 \times 10^{-4} \text{ rad s}^{-1}$). Is the effect larger or smaller?
3. For the rapidly rotating planet change *Nmu* to be 2000 and try varying *kmax* again. Do you notice any difference? What about the run time?

9 Conclusion

You should now have a basic grasp of how to use HERCULES. There are more options for modelling planets given in the user guide but these are likely to be used only by advanced users.

References

- W. B. Hubbard. High-precision Maclaurin-based models of rotating liquid planets. *The Astrophysical Journal*, 756(1):L15, 2012. ISSN 2041-8205. doi: 10.1088/2041-8205/756/1/L15. URL <http://iopscience.iop.org/article/10.1088/2041-8205/756/1/L15>.
- W.B. Hubbard, G. Schubert, D. Kong, and K. Zhang. On the convergence of the theory of figures. *Icarus*, 242:138–141, 2014. ISSN 00191035. doi: 10.1016/j.icarus.2014.08.014. URL <http://www.sciencedirect.com/science/article/pii/S001910351400428X>.
- WB B. Hubbard. Concentric Maclaurin spheroid models of rotating liquid planets. *The Astrophysical Journal*, 768(1):43, 2013. ISSN 0004-637X. doi: 10.1088/0004-637X/768/1/43. URL <http://iopscience.iop.org/article/10.1088/0004-637X/768/1/43><http://iopscience.iop.org/0004-637X/768/1/43>.
- Dali Kong, Keke Zhang, and Gerald Schubert. On the gravitational fields of Maclaurin spheroid models of rotating fluid planets. *The Astrophysical Journal*, 764(1):67, 2013. ISSN 0004-637X. doi: 10.1088/0004-637X/764/1/67. URL <http://iopscience.iop.org/article/10.1088/0004-637X/764/1/67>.