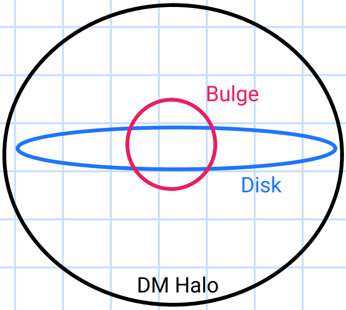
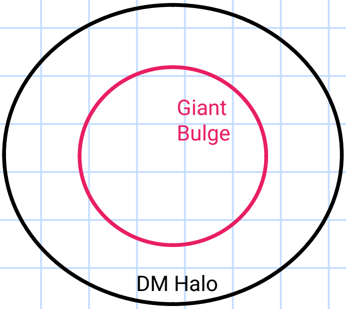
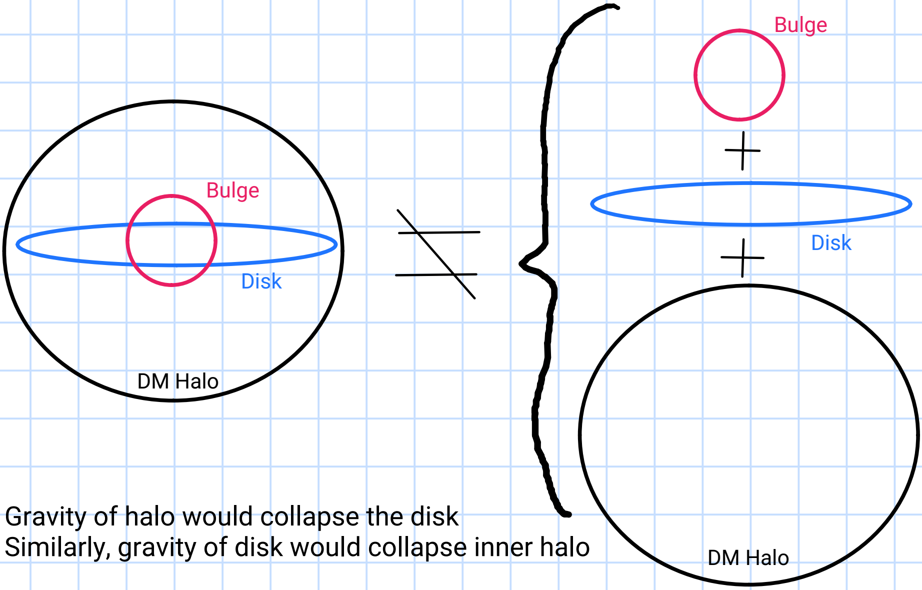
Simple multiple component galaxy models:


Disk galaxy model


Elliptical galaxy model

Typical density profile choices:
- DM halo: NFW profile
- Disk: exponential disk model
- Bulge: de Vacouleurs profile

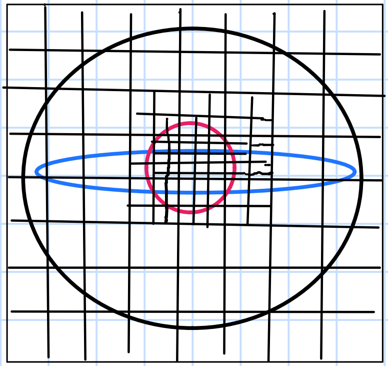
Cannot simply combine individual components:



Gravity of halo would collapse the disk
Similarly, gravity of disk would collapse inner halo

DICE : Disk Initial Conditions Environment

- Allows building of multiple component models
- Gravitational potential of combined components solved on a multilevel grid



Three levels:
High, medium, low
resolution

- Dynamical equilibrium found for each component based on gravitational potential of total model
- Can generate two galaxies simultaneously for modelling galaxy interactions
- Outputs a direct access binary file in Gadget format.

Toomre disk stability criteria:

Stellar disks have an internal velocity dispersion that helps to keep them stable against local gravitational instabilities. The Toomre criteria is used to choose the necessary velocity dispersion.

Radial velocity dispersion

$$Q_{toomre} = \frac{\kappa \sigma_r}{3.36 G \Sigma} > \sim 1.25 - 1.5$$

Epicyclic frequency

Disk surface density

A minimum dispersion at each radius is required to meet the criteria

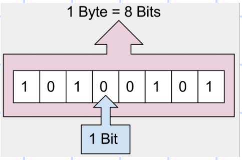
The epicyclic frequency can be approximated using:

$$\kappa = \frac{3}{R}\frac{\partial \phi}{\partial R} + \frac{\partial^2 \phi}{\partial R^2}$$

∴ High surface density & low potential gradients make the disk less stable

Direct access binary files

Binary files are often used in simulations because they are very compact and efficient use of disk space when particle numbers may be huge



- 4 byte integer = 32 bits, 2ˆ32 -1 numbers stored (0 to 4,294,967,295)
  With bit for negative, 2ˆ31 - 1: (-2,147,463,467 to + 2,147,463,467)
- 4 byte real number, 8 bits for exponent, 23 bits for mantissa + 1 bit sign, (m x 2ˆe, e.g., 2.03e24): (typical range: 1e-38 to 1e38, 7 decimal precision)
- 8 byte double precision number: (11 bits exponent, 52 mantissa + 1 bit sign: (typical range: 1e-308 to 1e+308, 15 decimal precision)

Writing a file:

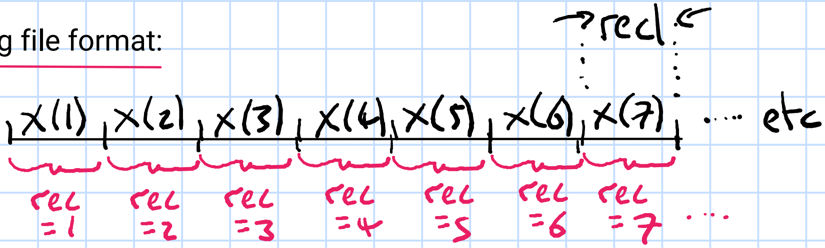
```
recl = 8
Open binary file, access=direct
Do i = 1, Nptcl
  | write(rec=i) X(i)
Enddo
```

First choose number if bytes for each record (double precision!)

Open file in direct access mode

Outputs vector X to each record of the file

Resulting file format:



Reading a file:

- With direct access, they are also very quick and efficient to read out data

```
Read(rec = 6) value
```

Immediately reads record number 6 (X(6)in the example above)

- There is no need to read all the data before, record by record. We can immediately extract the data we want (unlike for an ascii text file)
- Only problem is you cannot open the file in a text reader. You need a program to read the binary file. e.g., readgadgetfile.ipynb

'gf' (galaxy formation): an N-body Treecode for modelling galaxies

File conversion:

gf reads in a binary file of initial conditions (particle positions, velocities, masses). To use the initial conditions created by DICE, we will need to (1) convert from GADGET format binary to an ASCII file (so we can see the results in Topcat), and (2) convert the ASCII file to a gf binary format

Conversion steps:

1) DICE:	Produces GADGET binary
2) readgadgetfile.ipynb:	Converts GADGET binary (.g1) to ASCII file (.dat)
3) Topcat:	Reads ASCII file for visualisation
	Part 1
4) ascii2drt.f	Converts ASCII file to gf IC binary (gr_drk0000.drt)
5) gf:	Reads IC binary and runs the simulation, produces new output binary file (gf_drk.drt)
6) snapread.f	Reads output binary and writes out ASCII snapshot information

How gf works:

- like the python code from the star cluster project, gf is an N-body code that calculates accelerations between particles

- it is also a G=1 code (Msim=0.6e12 Msol, Lsim=12.42 kpc)

- similarly, it uses gravitational softening to avoid unphysical scattering between particles

- however, different from the python code, it uses a Treecode (see next page) to significantly speed up the gravitational calculations when there are many particles

- gf can also model gas in galaxies (it is a hydrodynamic code). But in this project, we will model only the stars and dark matter of a galaxy collision (see next project on cluster mergers for a hydrodynamic simulation)

What is a Treecode?

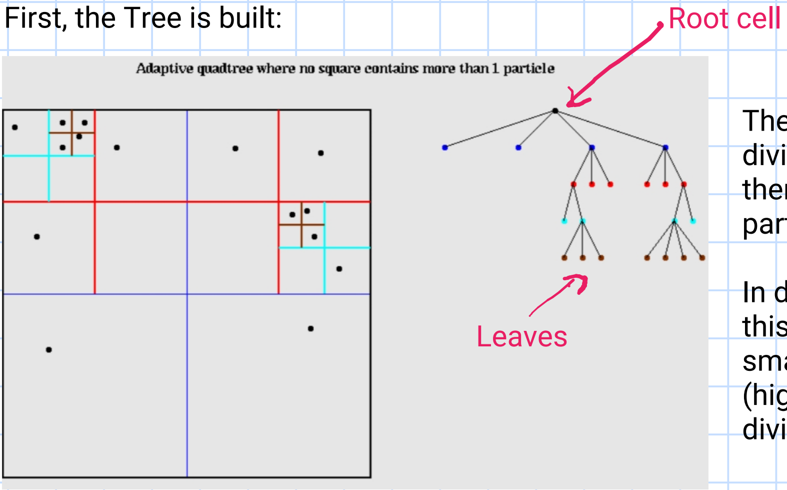
Efficient way to compute gravitational accelerations in high N systems:

In normal N-body code, the number of calculations scales as N^2
In Treecode, the number of calculations scales as N log N

N = 1000	N log N = 3000, N^2= 1 million, N^2/N log N = 333.3
N = 1 million	N log N = 6 million, N^2= 1e12, N^2/N log N = 166666.6

How does it work?

First, the Tree is built:



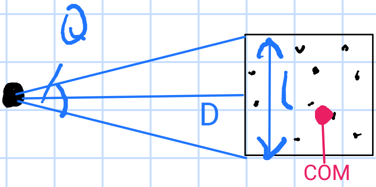
The 3D volume is divided up until there is only 1 particle per cell

In dense regions this requires smaller cells (higher level of division of root cell)

Now the code uses the Tree to approximate long distance gravitational forces.

Idea: exact particle distribution at long distances won't affect the acceleration much

Criteria based on opening angle:



$$\theta = \frac{L}{D}$$

← cell length

← distance

$\theta_{crit} \sim 0.5$ (typically)

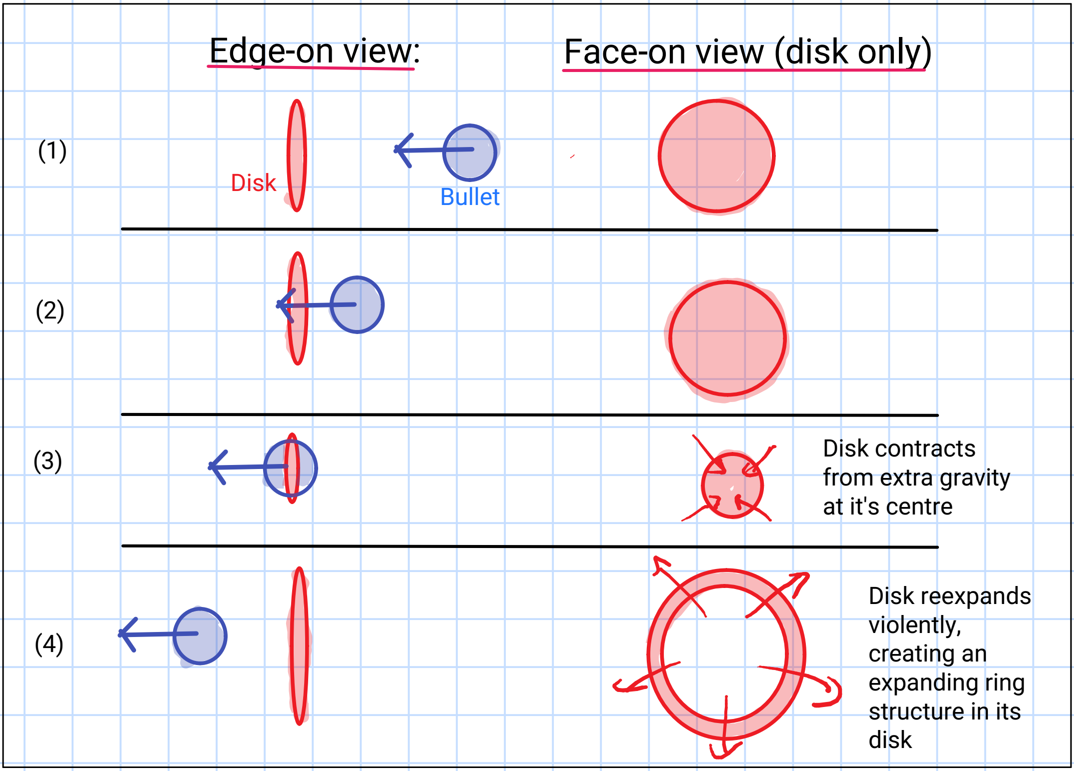
If $\theta < \theta_{crit}$, nearby: do full N-body calculation (expensive)

If $\theta > \theta_{crit}$, long distance: combine all particles in cell to point at COM. Just measure acceleration from that point (cheap)

What is a Collisional Ring Galaxy (CRG)?

The "Cartwheel" galaxy

CRGs when a spiral galaxy suffers a near head-on collision



Disk contracts from extra gravity at it's centre

Disk reexpands violently, creating an expanding ring structure in its disk

What you must do:

Part 1: Build an isolated disk galaxy model and analyse

(i) Download and install DICE. Read the documentation carefully and learn to use it (there are examples in dice/examples).

(ii) Use the code to build a disk galaxy with Mvir=5.5e11 Msol:
-NFW DM halo (50,000 DM particles), Mdm=85% total, c=12, rmax=170 kpc
-exponential stellar disk (20,000 star particles), 10% total, rexp=13 kpc, rmax=26 kpc, flatness=0.05, Toomre parameter>1.5.
-hernquist bulge (10,000 star particles), 5% total, rhern=6.5 kpc, rmax=13 kpc

(iii) Convert the GADGET file to ASCII format for analysis using readGadgetfile.ipynb. Use Topcat to make plots of the DM, disk and bulge components, seen edge-on and face-on. Add velocity vectors to the face-on disk image to reveal the rotation. Finally, make a plot of the rotation curve of the disk and explain how it was measured.

Part 2: Run the isolated disk simulation and test stability

(i) gf uses G=1 units with Msim=0.6e12 Msol, and Lsim=12.42 kpc. What is Vsim (in km/s) and Tsim (in Gyr) in this unit system?
Open the simulation parameters file (gi_par.asc). How long will the simulation run for (in Gyr) with the given parameters? What is the softening length (in kpc)?

(ii) Convert your ascii file to a set of gf binary files (e.g. gr_str0000.drt, gr_ gas0000.drt & gr_drk0000.drt). Now run your disk galaxy model simulation. gf outputs the snapshots to a new set of binary files (gf_drk.drt, gf_str.drt). Output the positions of the dark matter particles in each snapshot to an ascii file using the snapshot reader, and use your Lagrangian radii program from project 2 to measure the DM halo and stellar disk stability (separately).

Part 3: Simulate Collisional ring galaxy formation

(i) Use DICE to build the disk galaxy model and an elliptical galaxy combined:
Elliptical galaxy properties:
-Hernquist bulge (10,000 DM particles), Mhern=3.8e11 Msol, scalelength=3.0 kpc, rlim=100 kpc
Setup: disk galaxy at origin, elliptical galaxy at (0,0,100) kpc, with (0,0,-500 km/s)
What is the mass ratio of this merger?
Simulate their interaction, and make plots of the time evolution of the two DM halos (in the x-z plane) and the collisional ring (in the x-y plane).

(ii) What happens to the ring morphology if the collision is off axis? Run an additional simulation where the elliptical galaxy has an initial velocity of (0,50,-500) km/s, and make plots of the ring appearance with time and describe the differences.