
Simulações de N -corpos em Dinâmica Galáctica

Rubens Machado • rgmachado@gmail.com

Elaborado para a disciplina AGA0299 (Profª. Paula Coelho, IAG/USP)

Abril 2015

Neste exercício, utilizaremos as ferramentas do pacote NEMO para executar uma simulação de N -corpos de Dinâmica Galáctica. Criaremos as condições iniciais de uma galáxia composta de um disco estelar e de um halo de matéria escura. Com isso, poderemos executar a simulação, visualizar o resultado e interpretar a evolução dinâmica.

1 Condições iniciais

Para simular, por exemplo, a evolução gravitacional de uma galáxia ao longo do tempo, precisamos de condições iniciais.

Uma galáxia pode ser representada por uma distribuição de N partículas. Em última análise, as condições iniciais são simplesmente uma tabela especificando as coordenadas iniciais de posição (x, y, z) e de velocidade (v_x, v_y, v_z) de cada uma das N partículas, além da massa m de cada partícula. Trata-se portanto de uma tabela de 7 colunas e N linhas.

Assim como o disco estelar, o halo de matéria escura também pode ser representado por partículas. Evidentemente, o termo *partículas* aqui não tem nenhuma relação com as partículas elementares da Física. Falamos em partículas no contexto de massas puntiformes, do ponto de vista meramente computacional. Isto é, as partículas servem para *discretizar a massa*, para fins de cálculo. Um halo de matéria escura tem uma distribuição de massa (presumivelmente) contínua. Mas o representamos por um conjunto de partículas de massa m . Quanto maior o número de partículas, mais contínua tenderá a ser a aproximação. Similarmente, o disco estelar de uma galáxia tem uma certa distribuição de massa, que representamos por partículas. Entretanto, cada partícula *não* representa 1 estrela. Por exemplo, se uma galáxia de $M = 10^{11}M_{\odot}$ é representada por $N = 10^6$ partículas, então cada partícula corresponde a um *elemento de massa* com $m = M/N = 10^5M_{\odot}$.

Portanto, para criar condições iniciais precisamos especificar as posições, as velocidades e as massas de cada partícula. Em geral, trabalha-se com partículas de massas iguais, por simplicidade. As posições das partículas são criadas de tal modo que satisfaçam os perfis de densidade desejados. No caso de galáxias espirais, o perfil radial de densidade é bem representado pelo perfil exponencial:

$$\rho(R) = \rho_0 e^{-R/R_0} \quad (1)$$

onde ρ_0 é a densidade central e R_0 é o comprimento de escala (scale length) do disco, que no caso da Galáxia vale cerca de 3.5 kpc. Já a estrutura vertical do disco estelar como um todo costuma ser representada por uma dependência funcional como e^{-z/z_0} ou $\text{sech}^2(z/z_0)$, com uma escala de altura típica de algumas centenas de parsecs.

As galáxias estão imersas em halos aproximadamente esféricos de matéria escura. De acordo com as simulações cosmológicas, o perfil de densidade da matéria escura aparenta ser bem representados pelo chamado perfil NFW¹ que é dado por:

$$\rho(r) = \frac{\rho_s}{r/r_s (1 + r/r_s)^2}, \quad (2)$$

Mas aqui, vamos empregar um perfil similar, que é conveniente para simulações, chamado perfil de Dehnen²:

$$\rho(r) = \frac{(3 - \gamma) M_h}{4\pi} \frac{r_h}{r^\gamma (r + r_g)^{4-\gamma}}, \quad (3)$$

onde M_h é a massa total do halo e r_h é um comprimento de escala. O parâmetro γ controla a inclinação da região central do perfil $\rho(r)$. Por exemplo, com $\gamma \rightarrow 3$, temos um perfil com inclinação central mais acentuada, enquanto com $\gamma = 0$ temos densidade constante na região central (ou seja, um *core*).

2 Evolução gravitacional

Tendo as condições iniciais (i.e. posições e velocidades de cada partícula no instante $t = 0$), desejamos calcular a evolução do sistema ao longo do tempo. Para o caso particular $N = 2$, sabemos calcular as soluções analíticas. Para grandes N , no entanto, precisamos de soluções numéricas. E desejamos N tão grande quanto possível, para ter boa resolução em massa.

Do ponto de vista numérico, o problema gravitacional de N -corpos é conceitualmente simples. Dada uma distribuição de partículas, é trivial calcular a força resultante a que cada partícula está submetida. Esta resultante nada mais é do que a soma (vetorial) das forças exercidas sobre ela por cada uma das outras $N - 1$ partículas. Computacionalmente, para obter cada força basta medir as distâncias entre as partículas. Esse procedimento pode ser repetido para todas as N partículas, e agora sabemos a aceleração em cada uma delas (naquele instante t). Isso nos permite avançar no tempo de t para $t + dt$. E assim por diante. Este procedimento de força bruta se chama *soma direta* (direct summation) e infelizmente é inviável na prática, para os grandes N de que temos necessidade, pois o tempo de processamento seria proibitivo.

Para contornar esta dificuldade, foram desenvolvidos algoritmos extraordinariamente eficientes de cálculo da evolução gravitacional. Estes códigos reduzem imensamente o número de operações que precisam ser efetuadas e assim o tempo de processamento torna-se viável. Do ponto de vista computacional, este é um problema muito bem conhecido, e de soluções muito robustas.

Tanto na soma direta, quanto nos algoritmos mais sofisticados, existe um detalhe importante na implementação do cálculo de forças. É o chamado *softening length* (ϵ), que precisa ser introduzido como um termo adicional na força gravitacional:

$$|\mathbf{F}_i| = G \sum_{j=1}^N \frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2} \quad (4)$$

Trata-se de um truque numérico. Este termo serve para atenuar artificialmente a intensidade da força gravitacional, evitando que a força divirja quando duas partículas estão muito próximas, ou seja, quando $|\mathbf{r}_i - \mathbf{r}_j| \rightarrow 0$. Se duas partículas sofrem um encontro muito próximo, a força gravitacional verdadeira seria excessivamente intensa, provocando um espalhamento de grande ângulo. Mas lembre-se de que as partículas não estão representando estrelas (de dimensões

¹ Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493

²Dehnen, W. 1993, MNRAS, 265, 250

desprezíveis) e sim *elementos de massa* com talvez $10^5 M_\odot$. No contexto de uma galáxia, um elemento de massa desse porte certamente tem extensão apreciável e não seria realista que ele sofresse uma deflexão como a do encontro de duas massas puntiformes. Daí a necessidade de se introduzir o softening length, que tem dimensão de comprimento, obviamente. Em simulações de N -corpos, este ϵ é normalmente entendido como sendo o limite mínimo de resolução espacial do problema. Isto é, não se pode afirmar nada fisicamente confiável sobre o que acontece em escalas menores que $\sim \epsilon$.

3 Unidades

O NEMO usa, por default, unidades arbitrárias em que a constante de gravitação é tomada como $G = 1$, de tal modo que podemos especificar o sistema de unidades por convenção. Aqui, vamos adotar as seguintes unidades de comprimento e massa:

$$[L] = 1 \text{ kpc} \quad (5)$$

$$[M] = 10^{11} M_\odot \quad (6)$$

de tal modo que as unidades de tempo e de velocidade resultam ser aproximadamente:

$$[t] \simeq 1.5 \text{ Myr} \quad (7)$$

$$[v] \simeq 656 \text{ km/s} \quad (8)$$

4 Atividades práticas

Vamos usar alguns programas disponíveis no NEMO para executar a seguinte seqüência de passos:

1. Criar o disco estelar
2. Criar o halo de matéria escura
3. Juntar o disco e o halo
4. Rodar a simulação
5. Visualizar o resultado

Presume-se que o pacote já tenha sido previamente instalado com sucesso. Antes de começar, é preciso abrir um terminal e entrar na **C shell** (caso o default seja **bash**, por exemplo):

```
csch
```

e em seguida inicializar o ambiente do NEMO com este comando:

```
source /home/guest/nemo_cvs/nemo_start
```

Com isso, os programas de que precisamos deverão estar disponíveis neste terminal.

4.1 Criar o disco estelar

O primeiro passo consiste em criar as condições iniciais para um disco estelar. Isto é feito com o programa `mkexpdisk`:

```
mkexpdisk disk 10000
```

onde `disk` é o nome do arquivo de saída e 10000 é o número de partículas. O disco assim criado pode ser convenientemente visualizado com o programa `glnemo2`:

```
glnemo2 disk all
```

onde `all` seleciona todas as partículas do arquivo. Note que é possível observar o disco de qualquer orientação, dar zoom, mudar as cores, o tamanho e a textura das partículas, etc.

Para verificar rapidamente quais são os parâmetros do programa `mkexpdisk`, use:

```
mkexpdisk help=h
```

ou então, RTFM³:

```
man mkexpdisk
```

Vamos criar agora um disco exponencial, por exemplo, com as seguintes características: massa de $10^{11}M_{\odot}$, comprimento de escala de 3.5 kpc aproximadamente, altura de escala de 200 pc. E vamos truncar o disco em 15 kpc. Usaremos $N = 10^4$ partículas para o disco. Antes, lembre-se de apagar o teste anterior, pois os programas do NEMO nunca sobre-escrevem arquivos de mesmo nome.

```
rm disk
mkexpdisk out=disk nbody=10000 mdisk=1 alpha=0.3 z0=0.2 rcut=15
```

onde $\alpha = 1/R_0$.

O arquivo `disk` é criado em um formato binário específico do NEMO. Você pode imprimir as posições e velocidades do disco:

```
snapprint disk
```

O output, no terminal, são seis colunas com (x, y, z, v_x, v_y, v_z) e N linhas. É possível salvar esta tabelas em um arquivo de texto para plotar mais tarde:

```
snapprint disk > disk.txt
```

4.2 Criar o halo de matéria escura

O procedimento para criar o halo de matéria escura é análogo, usando o programa `mkdehnen`. Crie um teste qualquer e visualize-o:

```
mkdehnen halo 10000 gamma=0
glnemo2 halo all
rm halo
```

Consulte os parâmetros do programa `mkdehnen` e leia o manual:

```
mkdehnen help=h
man mkdehnen
```

Agora vamos criar um halo, por exemplo, com as seguinte características: 10^4 partículas, $2 \times 10^{11}M_{\odot}$, $\gamma = 0$, comprimento de escala 4 kpc, truncado em 50 kpc.

```
mkdehnen out=halo nbody=10000 mass=2 gamma=0 r_s=4 rmax=50
glnemo2 halo all
```

³<http://xkcd.com/293/>

4.3 Juntar o disco e o halo

Agora vamos juntar o disco e o halo para compor uma galáxia. Ocorre que estas condições iniciais na verdade estarão consideravelmente fora de equilíbrio! O disco não foi criado na presença do potencial gravitacional do halo. Equivalentemente, o halo também não foi criado na presença da massa do disco. Quando a simulação começar, as partículas do disco, por exemplo, sentirão o efeito de um potencial gravitacional mais profundo, para o qual elas não estavam preparadas. Ou seja, suas velocidades estarão inicialmente incorretas. Veremos como o sistema reage a seguir. Por enquanto, para juntar as componentes:

```
snapstack in1=disk in2=halo out=galaxy.ic zerocm=t
```

Temos então um arquivo de condições iniciais com 2×10^4 partículas. O programa que calcula a evolução gravitacional não fará distinção entre partículas do halo e partículas do disco: todas as partículas estão sujeitas às mesmas interações gravitacionais.

Veja que criamos um halo de matéria escura com apenas o dobro da massa do disco estelar, o que não é muito realista. Mas, neste exemplo didático, vai servir para minimizar as instabilidades.

Q1. Qual seria a proporção cosmológicamente adequada entre matéria escura e matéria bariônica? Especificamente, qual deveria ser a massa de matéria escura em uma galáxia cujo disco estelar tenha $10^{11}M_{\odot}$?

4.4 Rodar a simulação

Finalmente, tendo um arquivo de condições iniciais `galaxy.ic`, podemos rodar a simulação usando um código chamado `gyrfalcON`. Este código é excepcionalmente veloz, mas é serial. Isto é, só roda em 1 processador de cada vez, diferentemente dos códigos paralelos.

Vamos rodar a simulação durante apenas 300 Myr, o que em unidades internas corresponde a 200 unidades de tempo. Usaremos um softening length de 100 pc. Além disso, precisamos definir o passo de tempo de integração, que é dado por:

$$\tau = \frac{1}{2^{k_{\max}}} \quad (9)$$

de modo que escolhendo o parâmetro `kmax=3`, por exemplo, cada 1 unidade de tempo será dividida em $2^3 = 8$ passos. Assim, o tamanho de cada passo será:

$$\tau = \frac{1}{2^3} = 0.125 \text{ unidade de tempo} \simeq 187\,000 \text{ yr} \quad (10)$$

Ou seja, durante a evolução, as novas posições e velocidades das partículas serão calculadas a cada passo de 187 000 yr. Quanto mais fino este passo, mais acuradas serão as órbitas (e mais demorado será o cálculo, obviamente).

Q2. Dada a velocidade orbital do Sol, qual seria a distância percorrida (um segmento de reta) a cada passo de 187 000 yr? Qual o período da órbita do Sol ao redor do centro da Galáxia? Em passos de 187 000 yr, esta órbita seria calculada em quantos pontos, aproximadamente?

O programa não precisa necessariamente imprimir os resultados a cada passo de tempo calculado. Poderíamos usar um step mais largo (que só afeta a frequência com a qual o output é escrito, mas não o cálculo). Mas usaremos `step=0`, que faz com que todos os passos calculados

sejam impressos, pois isso nos dá uma visualização mais suave no final. Ou seja, teremos $2^3 = 8$ passos a cada 1 unidade de tempo, dando $8 \times 200 = 1600$ passos impressos. Por este motivo, se o arquivo de condições iniciais tinha ~ 1 MB, o arquivo de saída `galaxy.out` deveria ter ~ 1.6 GB. Ocorre que, por peculiaridades de como os programas foram implementados, as condições iniciais são criadas em double precision, mas o output do `gyrfalcON` sai como float. Então na verdade, o arquivo de saída deverá ter ~ 800 MB.

Finalmente, o comando para rodar a simulação é:

```
gyrfalcON galaxy.ic galaxy.out kmax=3 eps=0.1 step=0 tstop=200
```

O tempo de processamento dependerá da velocidade do processador. Em um processador comum atual (e.g. Intel Core i5 de 1.7 GHz), este exemplo deverá ficar pronto em menos de 2 minutos, em condições ideais. Veja que o `gyrfalcON` imprime uma série de diagnósticos a cada passo de tempo. Consulte o manual:

```
gyrfalcON help=h  
man gyrfalcON
```

Em particular, a última coluna mostra o tempo total decorrido desde o início (wallclock time em hh:mm:ss.ss). A sexta coluna imprime uma quantidade interessante: $-2T/W$ onde T é a energia cinética total e W é a energia potencial total. De acordo com o Teorema do Virial, esta razão (a rigor, sua média temporal) deve ser igual a 1 para um sistema em equilíbrio dinâmico. Como nossas condições iniciais estão consideravelmente *fora de equilíbrio*, por construção, a razão do virial difere bastante da unidade no início, mas dentro de algum tempo começa a tender para 1 com boa precisão. É possível adicionar ao comando do `gyrfalcON` a opção:

```
logfile=galaxy.log
```

Assim, as 14 colunas de diagnóstico vão para um arquivo de texto, que você pode usar depois para plotar a evolução temporal das energias e da razão do virial, por exemplo.

Q3. Como se comportou a conservação de energia neste exemplo? Verifique também a conservação do momento angular e a velocidade do centro de massa. Estes diagnósticos nos informam sobre a acurácia do cálculo das forças gravitacionais. Isto basta para garantir o realismo astrofísico da simulação?

4.5 Visualizar a simulação

Para visualizar a evolução do disco, selecionamos as primeiras 10000 partículas (i.e. desde a partícula 0 até a partícula 9999):

```
glnemo2 galaxy.out select=0:9999
```

Observando a evolução temporal, note as graves instabilidades provocadas pela falta de equilíbrio das condições iniciais. Este resultado é *não-físico* no sentido de que tais movimentos não se dão nas galáxias do Universo real. Entretanto, as forças gravitacionais estão sendo calculadas corretamente, dada aquela configuração inicial implausível que fornecemos. Este é o princípio computacional conhecido como GIGO (Garbage In, Garbage Out)⁴.

⁴On two occasions I have been asked, “Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?” ... I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question. — Charles Babbage, Passages from the Life of a Philosopher

Passados os instantes iniciais, as partículas começam a se acomodar em órbitas mais estáveis. Na periferia do disco, algumas partículas acabam se perdendo. Mas na região central ($R \lesssim 5$ kpc) – onde a densidade é mais alta e portanto as escalas de tempo são mais curtas – a dinâmica é mais plausível. Inclusive, é possível observar a formação de uma barra.

Q4. É possível estimar visualmente o período de rotação da barra? Qual sua velocidade angular? Qual o comprimento da barra? O que seria necessário medir para se determinar o raio de co-rotação?

Equivalentemente, podemos visualizar a evolução do halo. Note que o halo não tem rotação: tem dispersão isotrópica de velocidades. Com um zoom na região central, é fácil ver que a dispersão de velocidades é bem mais alta no centro.

```
glnemo2 galaxy.out select=10000:19999
```

Q5. Esta galáxia tem disco estelar e halo de matéria escura. O que falta? Das componentes não incluídas neste tipo de simulação, quais você espera que tenham a maior relevância para a evolução dinâmica?

É possível imprimir em um arquivo de texto as coordenadas (x, y, z, v_x, v_y, v_z) de um dado instante de tempo do output. Por exemplo, do último instante:

```
snapprint galaxy.out times=200 > galaxy_200.txt
```

As 10000 linhas iniciais são o disco:

```
head -10000 galaxy_200.txt > disk_200.txt
```

E as 10000 linhas finais são o halo:

```
tail -10000 galaxy_200.txt > halo_200.txt
```

Com estas tabelas, você pode medir várias coisas: por exemplo, os perfis de densidade finais do disco e do halo, as dispersões de velocidade, a curva de rotação, etc. Também é possível fazer um script para imprimir em arquivos de texto vários instantes de tempo. Repetindo o procedimento, é simples gerar centenas de gráficos e fazer uma animação para ver como um dado perfil evolui no tempo.

Interpretações: Apesar de este exemplo didático servir para ilustrar alguns procedimentos e técnicas de simulações de N -corpos, não é prudente extrair dele interpretações astrofísicas muito rigorosas, por causa da baixa resolução e do não-equilíbrio das condições iniciais.

Q6. Colisões de galáxias: Por exemplo, criando duas galáxias iguais ou similares, é possível juntá-las com o `snapstack`, dispondo-as com uma separação espacial `deltar` e uma velocidade relativa `deltav`. O que acontece com discos que sofrem um encontro com um parâmetro de impacto? Quais outras interações entre galáxias poderiam ser bem estudadas com simulações de N -corpos?

5 Referências

- ***N*-corpos:**

N-body simulations

[http://www.scholarpedia.org/article/N-body_simulations_\(gravitational\)](http://www.scholarpedia.org/article/N-body_simulations_(gravitational))

The *N*-body problem

<http://sverre.com/>

The Art of Computational Science

<http://www.artcompsci.org/>

- **Alguns códigos:**

Gadget <http://www.mpa-garching.mpg.de/gadget/>

Arepo <http://www.mpa-garching.mpg.de/~volker/arepo/>

Ramses <http://www.ics.uzh.ch/~teyssier/ramses/RAMSES.html>

Flash <http://flash.uchicago.edu/site/flashcode/>

- **Algumas simulações cosmológicas:**

Illustris

<http://www.illustris-project.org/>

Millennium

<http://www.mpa-garching.mpg.de/galform/virgo/millennium/index.shtml>

Millennium-XXL

<http://galformod.mpa-garching.mpg.de/mxxlbrowser/>

Aquarius

<http://www.mpa-garching.mpg.de/aquarius/>

DEUS

<http://www.deus-consortium.org/>

Bolshoi

<http://hipacc.ucsc.edu/Bolshoi/>

- **Outros:**

<http://www.astro.iag.usp.br/~rgmachado/videos.html>

Apêndice A: Instalação do NEMO

O pacote NEMO - A Stellar Dynamics Toolbox⁵ pode ser instalado localmente, isto é, em um diretório qualquer do usuário, sem necessitar de permissões de root. Em princípio, a instalação se resume a este procedimento (antes de começar, entre na `ssh`):

```
wget ftp://ftp.astro.umd.edu/pub/nemo/nemo_cvs.tar.gz
tar xzf nemo_cvs.tar.gz
cd nemo_cvs
cvs update -d
chmod +x AAA_SOURCE_ME.csh
./AAA_SOURCE_ME.csh
```

Mas esta instalação depende da presença de diversas bibliotecas. Dificilmente, ela terá êxito nas primeiras tentativas. Consulte o arquivo `install.log` gerado em `nemo_cvs/` e tente instalar os pacotes necessários no seu sistema operacional (compiladores diversos, etc) até satisfazer as dependências. Estritamente, não é necessário que *todos* os programas do NEMO compilem sem erro. Para tentar recompilar explicitamente apenas um dado programa, use o comando `mknemo`. Por exemplo:

```
mknemo gyrfalcON
```

Para inicializar o ambiente do NEMO (sempre estando na `ssh`), use o comando:

```
source /home/guest/nemo_cvs/nemo_start
```

a cada vez que abrir um novo terminal. Você pode criar um alias, adicionando a linha:

```
alias nemo "source /home/guest/nemo_cvs/nemo_start"
```

no arquivo oculto `/home/guest/.cshrc` por exemplo. Agora, em terminais recém-abertos, o ambiente se inicializa simplesmente com o comando:

```
nemo
```

e se desfaz com o comando inverso:

```
omen
```

• Instalação do `glnemo2`

Caso a instalação do `glnemo2` não tenha sucesso, uma possibilidade é seguir as instruções na página⁶. Tendo compilado o `glnemo2` a partir do source disponível na página, copie o executável para o diretório `nemo_cvs/bin/`. Assim, ele estará disponível junto com os outros executáveis do NEMO quando o ambiente for inicializado. O bom funcionamento do `glnemo2` depende da presença de uma placa de vídeo decente. A propósito, o `glnemo2` também serve para visualizar simulações do `Gadget-2` e do `RAMSES`.

Alternativamente, pode-se seguir as instruções desse link:
<http://snatverk.blogspot.com.br/2013/02/how-to-install-glnemo2-in-ubuntu.html>

PS: Na parte installation há um erro de digitação: deve-se digitar `qmake -recursive` (em vez de " `qmake -revursive` " como mostrado no link). (Dica de André Luiz Figueiredo)

⁵<http://bima.astro.umd.edu/nemo/>

⁶<http://projets.lam.fr/projects/glnemo2>

Apêndice B: Uma simulação mais realista

O exemplo usado nas seções anteriores pode ser executado muito rapidamente – durante uma aula, por exemplo – e serve para ilustrar basicamente os procedimentos envolvidos em uma simulação de N -corpos, ao menos do ponto de vista funcional. No entanto, o resultado daquela simulação não dá margem a muitas interpretações físicas sobre a dinâmica da galáxia, pois ela sofre de pelo menos duas limitações graves: (i) a falta de equilíbrio das condições iniciais, por construção e (ii) a baixa resolução.

Poderíamos realizar uma simulação de mais alta resolução usando um outro programa do NEMO chamado `magalie`, que constrói as condições iniciais – em equilíbrio – de um sistema composto de disco + bojo + halo. Consulte os detalhes em Boily et al. (2000)⁷. Veja os parâmetros no manual:

```
magalie help=h
man magalie
```

Como, neste caso, o comprimento de escala do disco é necessariamente unitário (bem como a massa do disco), convém adotar outro sistema de unidades (também com $G = 1$). Convencionamos:

$$[L] = 3.5 \text{ kpc} \quad (11)$$

$$[M] = 5 \times 10^{10} M_{\odot} \quad (12)$$

$$[t] \simeq 14 \text{ Myr} \quad (13)$$

$$[v] \simeq 248 \text{ km/s} \quad (14)$$

Vamos criar um disco com $5 \times 10^{10} M_{\odot}$, comprimento de escala de 3.5 kpc, escala de altura 700 pc e 20000 partículas. O bojo terá apenas 10% da massa do disco e comprimento de escala de 1.75 kpc. Para que as partículas do disco e do bojo tenham a mesma massa, criamos o disco com 20000 partículas e o bojo com 2000. O halo terá $2.5 \times 10^{11} M_{\odot}$ e comprimento de escala de 7 kpc. Se desejássemos manter *todas* as partículas com a mesma massa, então o halo precisaria ter 100000 partículas. No entanto vamos empregar apenas 40000 partículas para o halo. Isto significa que o halo terá uma resolução em massa um pouco pior do que as componentes estelares, mas isto reduz o tempo de cálculo sem conseqüências temerárias para a morfologia do disco. A linha de comando para criar esta galáxia será:

```
magalie out=galaxy.ic ndisk=20000 hdisk=0.2 nbulge=2000 mbulge=0.1
rbulge=0.5 mhalo=5 rhalo=2 nhalo=40000 Qtoomre=1.0
```

onde usamos o parâmetro de Toomre $Q = 1$. Em linhas gerais, este parâmetro é proporcional à dispersão de velocidades e controla a estabilidade dinâmica do disco. Se $Q > 1$, o disco é dinamicamente “quente” e será estável contra perturbações, essencialmente. Se $Q < 1$, o disco é dinamicamente “frio” e tende a ser instável. Por exemplo, discos frios podem sucumbir mais facilmente às instabilidades que dão origem à barra. Em discos dinamicamente quentes, a barra (caso se forme) será mais tardia e/ou mais fraca. Tente refazer a simulação com $Q = 0.5$ e com $Q = 2$, por exemplo e observe as diferenças na formação da barra.

Agora vamos rodar estas condições iniciais com `kmax=6`. Cada unidade de tempo será dividida em 64, de modo que o passo de integração será:

$$\tau = \frac{1}{2^6} = 0.015625 \text{ unidade de tempo} \simeq 218 \text{ 750 yr} \quad (15)$$

Usando `tstop=800` unidades de tempo, poderemos seguir a evolução desta galáxia durante ~ 11.2 Gyr, ou seja, quase 1 tempo de Hubble. Desta vez, vamos imprimir os resultados a cada

⁷Boily, C., et al. 2001, *New Astronomy*, 6, 27

0.2 unidades de tempo, para evitar um arquivo de saída imenso. Ainda assim, o output terá algo como ~ 7 GB. Sempre estime o tamanho do arquivo antes de começar a simulação. O comando agora será:

```
gyrfalc0N galaxy.ic galaxy.out kmax=6 eps=0.05 step=0.2 tstop=800
```

Esta simulação deverá levar mais de 2 horas para terminar de rodar, em um processador típico atual. Como o código é serial, não adianta pretender rodá-la em um computador com vários processadores. O máximo que você pode fazer é rodar várias simulações diferentes *simultaneamente*, mas cada uma delas é independentemente limitada pela velocidade de 1 processador individual da máquina em questão. Tente aumentar todos os números de partículas por um fator 10 ou 100 e meça o tempo de processamento (extrapolando a partir do começo da simulação, usando a última coluna de diagnósticos).

Finalmente, para visualizar as componentes estelares (disco + bojo):

```
glnemo2 galaxy.out select=0:21999
```

