

CÓDIGOS PARA PREPARAÇÃO DE CONDIÇÕES INICIAIS EM SIMULAÇÕES DE COLISÕES DE GALÁXIAS

Dino Beghetto Junior^{1,2}, Irapuan Rodrigues de Oliveira Filho²

¹UNIFESP – Universidade Federal de São Paulo/ICT - Instituto de Ciência e Tecnologia, Rua Talim, 330 - Vila Nair - São José dos Campos-SP, dino.beghetto@unifesp.br

²UNIVAP/IPD – Instituto de Pesquisa e Desenvolvimento, Av. Shishima Hifumi, 2911 - Urbanova - São José dos Campos-SP

Resumo- Atualmente a astronomia caminha juntamente com a computação, sendo esta uma ferramenta indispensável no estudo e compreensão dos diversos processos físicos envolvidos em grandes sistemas estelares, pois permite a realização de simulações numéricas para o estudo de problemas que, de outra forma, se tornariam praticamente insolúveis. No caso específico de dinâmica de galáxias, seu estudo seria impossível sem o auxílio de simulações numéricas, pois a escala de tempo de tais fenômenos é muito grande para se obter informações somente através de observações. Tendo isso em vista, esse trabalho discute alguns códigos desenvolvidos em linguagem C, que visam melhorias na preparação de condições iniciais para simulações, apresentando os resultados obtidos e fazendo comparações com outro código que já vinha sendo utilizado pelo grupo de pesquisa.

Palavras-chave: Astrofísica extragaláctica, simulações de N-Corpos, galáxias em interação.

Área do Conhecimento: Ciências Exatas e da Terra

Introdução

Os encontros de sistemas estelares, como aglomerados de estrelas, galáxias e aglomerados de galáxias, têm modificado substancialmente as estruturas cósmicas ao longo de toda a evolução do Universo, e as simulações numéricas auxiliam na obtenção de uma melhor compreensão das interações entre esses sistemas. A possibilidade de simular interações com diversas condições iniciais permitiu entender estruturas antes observadas como bizarras, não compreensíveis em sistemas estáveis, como fenômenos provenientes de diversos tipos de interação. Simulações podem testar idéias teóricas, bem como melhorar a compreensão sobre o funcionamento dos sistemas reais (RODRIGUES, 1999). Neste contexto, o presente trabalho apresenta alguns códigos, escritos em linguagem C, que auxiliam na preparação dos modelos de galáxias para simulação numérica.

O estudo de colisões de galáxias feito pelo grupo de pesquisa coordenado pelo Dr. Irapuan Rodrigues, do IPD-UNIVAP, inicia-se com a obtenção de dados observacionais nos grandes telescópios através de imageamento e espectroscopia (2D e 3D) óptica e IR e eventualmente, com o mapeamento da quantidade e campo de velocidades do hidrogênio neutro, de onde se faz um estudo morfológico/químico/cinemático dos sistemas de interesse. Numa segunda fase realiza-se o modelamento numérico dos sistemas com o objetivo de se estudar sua história dinâmica,

sendo a ferramenta principal o GADGET-2, um código de alto desempenho para simulações de N-Corpos especialmente escrito para ser executado em computadores paralelos (SPRINGEL, 2005b).

É nessa última etapa que se encaixa nosso trabalho. O fluxograma a seguir (Figura 1) indica os passos normalmente executados para preparação e disparo das simulações numéricas de colisões de galáxias.

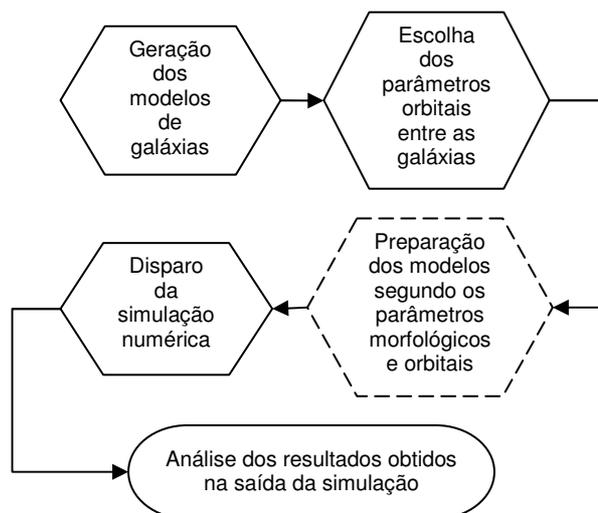


Figura 1 - Fluxograma dos passos para simulação

Após a geração dos modelos, a decisão dos parâmetros orbitais entre as galáxias é feita, pois a observação não fornece todos os parâmetros necessários. Na fase seguinte, destacada no

fluxograma (Figura 1) com a caixa em linha tracejada, os modelos são preparados segundo suas morfologias e seus parâmetros orbitais, momento onde os códigos a serem discutidos neste artigo se encaixam. A saber, são três códigos, desenvolvidos em ambiente Linux, baseados no código base de exemplo escrito pelo orientador. Posteriormente é feito o disparo da simulação, quando o sistema estudado é evoluído no tempo. Finalmente analisam-se os resultados obtidos, podendo-se estudar a dinâmica do sistema simulado juntamente com dados obtidos observacionalmente.

Metodologia

Os três programas apresentados a seguir foram baseados no código base de exemplo, escrito pelo orientador, de nome “gadget_read_write”, que lê um arquivo de snapshot no formato estruturado do GADGET-2 (Tabela 1) e escreve de volta os dados. Suas chamadas para execução são feitas via linha de comando (os programas não possuem interface gráfica).

Um dos programas se chama “gadget_galrunpars”, que lê um snapshot ou arquivo de condições iniciais, faz os cálculos necessários de maneira separada para cada um dos seis tipos de partícula definidos pelo GADGET-2 (a saber: gás, halo, disco, bojo, matéria escura e buracos negros) e também para o sistema como um todo. A saída do programa apresenta informações fundamentais para o preparo de simulações, como raio que contém 10%, 50% e 90% da massa, separações médias entre partículas para cálculo do *softening parameter*, *crossing times*, etc.

O próximo programa, que foi batizado como “gadget_desloca”, possui uma rotina que recentra o modelo em um sistema de coordenadas, outra rotina que desloca as posições de cada partícula, e uma terceira que desloca suas velocidades. Tais funções são acionadas por *flags*, de acordo com o que o usuário digita como parâmetro de entrada na linha de comando para chamada do programa, que possui um breve Guia de Usuário para mostrar como fazer as chamadas de cada função, assim como, caso acionadas, indicar o quanto serão deslocadas as partículas em velocidade e/ou em posição.

Tabela 1 – Blocos de saída definidos nos arquivos de formato do GADGET-2 (retirada de SPRINGEL, 2005a)

#	Block ID	HDF5 identifier	Block Content
1	HEAD		Header
2	POS	Coordinates	Positions
3	VEL	Velocities	Velocities
4	ID	ParticlesIDs	Particle ID's
5	Mass	Masses	Masses (only for particle types with variable masses)
6	U	Internal Energy	Internal Energy per unit mass (only SPH particles)
7	RHO	Density	Density (only SPH particles)
8	HSML	SmoothingLength	SPH smoothing length h (only SPH particles)
9	POT	Potential	Gravitational potential of particles (only when enable in makefile)
10	ACCE	Acceleration	Acceleration of particles (only when enable in makefile)
11	ENDT	RateOfChangeOfEntropy	Rate of change of entropy function of SPH particles (only when enable in makefile)
12	TSTP	TimeStep	TimeStep of particles (only when enable in makefile)

Posteriormente outra rotina foi criada e adicionada ao programa que recentra e desloca o modelo, que foi rebatizado como “gadget_tilt”; essa última atualização, além das funções antigas, inclina o modelo de acordo com as coordenadas cartesianas do vetor de *spin* (que é um vetor paralelo ao vetor momento angular de um modelo de galáxia espiral) dadas pelo usuário.

O Guia de Usuário, citado quando falamos do gadget_desloca, também aparece no gadget_tilt, no entanto atualizado com a informação sobre como chamar a rotina de inclinação do modelo.

Tais programas são de importância para o grupo de pesquisa, pois aceleram o processo de preparação de condições iniciais; os programas utilizados previamente pelo grupo fazem duas conversões de dados, pois eram escritos para um formato diferente, enquanto os códigos discutidos nesse trabalho não precisam fazer nenhuma conversão, trabalham com arquivos binários.

Resultados

Um exemplo de chamada em linha de comando e saída correspondente na tela para o `gadget_galrunpars` é dado na Figura 2. Por motivo de extensão da saída, esta não será apresentada na íntegra, mas somente trechos mais relevantes, como os dados para o tipo "0" de partícula, que é o tipo de partículas que representa o gás no sistema (deixamos claro aqui, no entanto, que a saída para cada tipo de partícula do sistema possui formato similar).

Apresenta-se um exemplo de chamada do `gadget_desloca` por linha de comando com acionamento de certas *flags* na Figura 3, juntamente com a saída em tela. Note que foi chamada uma rotina para deslocar o modelo em posição e outra para deslocar em velocidade, ambas seguidas das coordenadas desejadas, mas também é possível chamar somente uma delas, deslocando somente em velocidade ou em posição.

A Figura 4 mostra que a chamada do programa `gadget_tilt` sem nenhum parâmetro abre o Guia de Usuário em tela para consulta.

Um exemplo de chamada do `gadget_tilt` de forma similar ao feito no exemplo da Figura 3 é apresentado na Figura 5, além de um trecho da saída de tela. Nesse caso, somente a rotina para inclinar o modelo foi chamada, com coordenadas $\langle 1, 1, 1 \rangle$ do vetor de *spin*, o que gera um ângulo de inclinação de aproximadamente 55 graus e um ângulo de posição de 45 graus, ambos calculados e mostrados na tela para o usuário.

Novamente por motivo de extensão, como foi explicitado quando citamos previamente a Figura 2, as saídas de tela mostradas nas Figuras 3 e 5 também são trechos, e não as saídas totais criadas pelos programas. Entretanto, acreditamos que, da forma que estão apresentadas, as figuras constituem um ótimo exemplo e não comprometem o entendimento do artigo.

```
dino@linux-iso8:~/simulation/conversion/gadget_tools>./gadget_galrunpars -o GADGET-IC_123_teste GADGET-IC_123

Computing data to output file (gadget format)...

Total Number of particles = 25600
=====
Maximun radius = 5.699341
Average velocity = 3.753912
Total mass = 0.285970
=====
1/10 of the mass: R10 = 3.116601
                   <V10>= 3.757368
1/2  of the mass: R50 = 3.798010
                   <V50>= 3.755156
9/10 of the mass: R90 = 4.624808
                   <V90>= 3.754254
=====
Mean particle separation:
inside R10: 0.479623,
inside R50: 0.321136,
inside R90: 0.322573.
=====
Crossing times:
Using R10 and <v10>: Tcross = 0.829464.
Using R50 and <v50>: Tcross = 1.011412.
=====
Relaxation time: 2550.863281
=====

Type = 0 - Number of particles = 8192
=====
Maximun radius = 5.526233
Average velocity = 3.753436
Total mass = 0.009360
=====
1/10 of the mass: R10 = 3.138489
                   <V10>= 3.758823
1/2  of the mass: R50 = 3.788907
                   <V50>= 3.753848
9/10 of the mass: R90 = 4.516289
                   <V90>= 3.752794
=====
Mean particle separation:
inside R10: 0.670897,
inside R50: 0.473652,
inside R90: 0.464126.
=====
Crossing times:
Using R10 and <v10>: Tcross = 0.834966.
Using R50 and <v50>: Tcross = 1.009340.
=====
Relaxation time: 917.510986
=====
```

Figura 2 – chamada por linha de comando e trecho da saída de tela do `gadget_galrunpars`

```
dino@linux-iso8:~/simulation/conversion/gadget_tools>./gadget_desloca -o GADGET-IC_123_teste -spos 6 5 4 -svel 7 8 9 GADGET-IC_123
reading `GADGET-IC_123' ...

By default in GADGET format, groups are:
  Npart group 0 = 8192 --> assuming as GAS
  Npart group 1 = 8192 --> assuming as DARK MATTER (halo)
  Npart group 2 = 8192 --> assuming as STARS (disk)
  Npart group 3 = 1024 --> assuming as STARS (bulge)
  Npart group 4 = 0 --> assuming as STARS (other)
  Npart group 5 = 0 --> assuming as BLACK HOLE

allocating memory...
allocating memory...done

reordering....
reordering...done

reordering data by type...
reordering data by type...done

writing data to output file (gadget format)...
Position (Original Mass Center): 1.00022 3.00067 2.00044
Velocity (Original Mass Center): 1.00022 3.00067 2.00044

writing binary format file...
  Ngas = 8192
  Nhalo = 8192
  Ndisk = 8192
  Nbulge = 1024
  Nstars4 = 0
  Nbh5 = 0
writing binary format file... done
writing data to output file (gadget format)...done

wrote file: GADGET-IC_123_teste
```

Figura 3 – chamada do gadget_desloca com acionamento de flags para deslocar o modelo em posições em coordenadas <6,5,4> e em velocidades <7,8,9>, e saída de tela

```
dino@linux-iso8:~/simulation/conversion/gadget_tools>./gadget_tilt

USAGE: ./gadget_tilt INPUT_FILE [options]
OPTIONS INCLUDE:
  -pot          :turn on potential outputs
  -nr          :turn off recentering
  -spos XX YY ZZ      :shifting positions by XX YY ZZ
  -svel VXX VYY VZZ  :shifting velocities by VXX VYY VZZ
  -tt          :tilt by new spin direction's coordinates SX SY SZ
  -o FILE       :sets output_file to FILE
```

Figura 4 – chamada do gadget_tilt sem parâmetros para visualização do User Guide

```
dino@linux-iso8:~/simulation/conversion/gadget_tools>./gadget_tilt -o GADGET-IC_123_teste -tt 1 1 1 GADGET-IC_123

writing data to output file (gadget format)...
  Positions (Original Mass Center): 1.00022 3.00067 2.00044
  Velocities (Original Mass Center): 1.00022 3.00067 2.00044
=====

Inclina um modelo de galáxia dada a nova direcao do vetor de spin, que inicialmente supoe-se estar apontando para (0,0,1)

Condicoes iniciais:
-> Nome do arquivo de entrada e de saida do modelo;
-> direcao do novo vetor de spin

=====

I (inclination) = 54.735609
W (position) = 45.000001

writing binary format file...
  Ngas = 8192
  Nhalo = 8192
  Ndisk = 8192
  Nbulge = 1024
  Nstars4 = 0
  Nbh5 = 0
writing binary format file... done
writing data to output file (gadget format)...done

wrote file: GADGET-IC_123_teste
```

Figura 5 – chamada do gadget_tilt com acionamento de flag para inclinar o modelo de acordo com coordenadas do vetor de spin <1,1,1>, e trecho da saída de tela

Discussão

Depois de os programas serem executados, os modelos tratados podem ser observados através da ferramenta Gadgetviewer, que lê os arquivos de saída e cria uma visualização do modelo.

A Figura 6 mostra o modelo de galáxia antes de passar pelos tratamentos de deslocamento e inclinação, respectivamente com visualização no plano XY (a) e no plano XZ (b) em um sistema de coordenadas cartesianas; o modelo após executado pelos programas gadget_desloca e gadget_tilt (ambos feitos com chamadas como nos exemplos apresentados nas Figuras 3 e 5, porém para um modelo com um milhão de partículas, para melhor visualização) é apresentado na Figura 7, de maneira similar à feita na Figura 6.

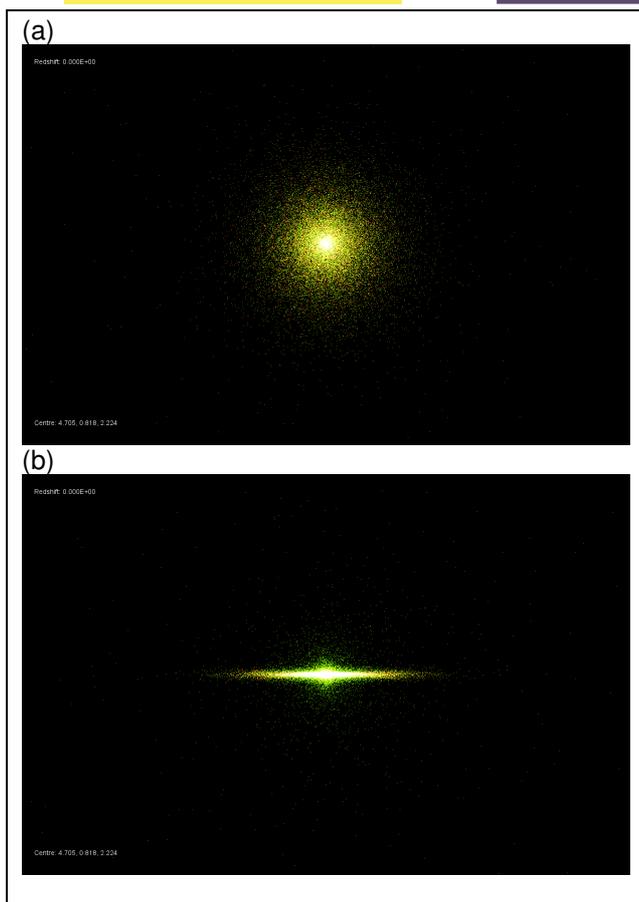


Figura 6 – Visualização pelo Gadgetviewer da projeção nos planos XY (a) e XZ (b) do modelo original da galáxia

Conclusão

Os códigos aqui discutidos mostram claramente sua utilidade para o grupo de pesquisa, pois permitem uma aceleração no processo de preparação de condições iniciais para simulação de galáxias, alcançando os resultados esperados.

Os modelos foram recentrados, deslocados e inclinados de forma correta, e visualizados pelo Gadgetviewer de maneira bastante clara. Com esses resultados satisfatórios, é possível pensar em códigos que auxiliem de forma similar em outras partes do processo da simulação, como, por exemplo, a colisão propriamente entre dois modelos tratados.

Todo o esforço é revertido em uma melhor compreensão da evolução dinâmica do sistema em estudo, que não pode ser alcançado senão com o auxílio de simulações numéricas.

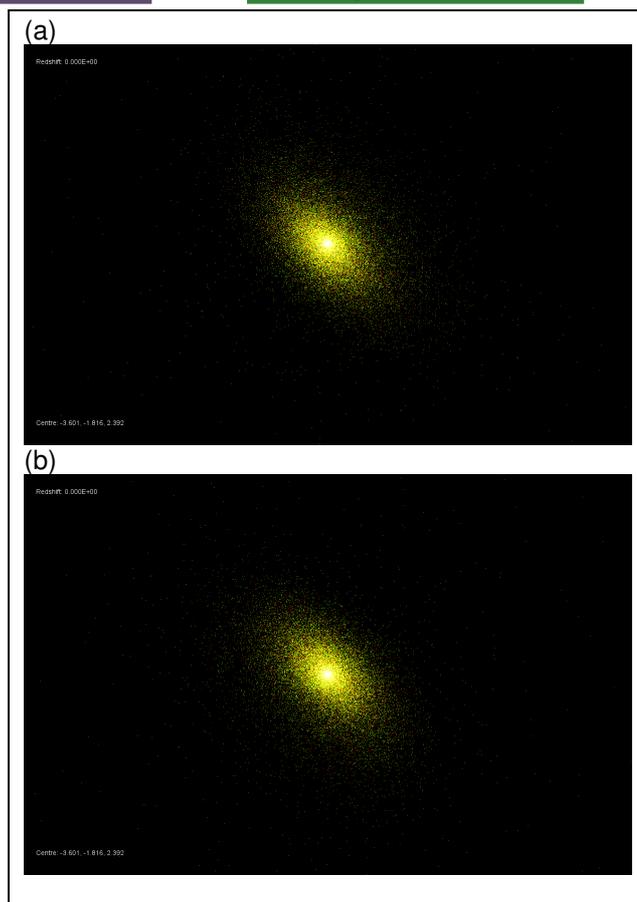


Figura 7 – Visualização pelo Gadgetviewer da projeção nos planos XY (a) e XZ (b) do modelo da galáxia inclinado pelo gadget_tilt

Agradecimentos

À FAPESP, processo número 2010/12969-8, pelo apoio financeiro.

Referências

- SPRINGEL, Volker. User guide for GADGET-2. Max-Planck Institute for Astrophysics, Garching, Germany. 2005a.
- SPRINGEL, Volker. The cosmological simulation code GADGET-2. Monthly Notices of the Royal Astronomical Society, vol. 364, Issue 4, pp. 1105-1134. 2005b.
- RODRIGUES, I. Interações de Sistemas Estelares. 1999. Tese (Doutorado em Ciências) – Instituto de Física, Universidade Federal do Rio Grande do Sul, 1999.