

Taller #6. Métodos computacionales / FISI 2028
Semestre 2014-I.

Profesor: Jaime E. Forero Romero

Abril 8, 2014

Esta tarea debe resolverse por parejas (i.e. grupos de 2 personas) y debe estar en un repositorio de la cuenta de github de uno de los miembros de cada equipo con un commit final hecho antes del medio día del martes 22 de Abril del 2013. El esqueleto de todos los códigos debe estar en el repositorio antes del medio día del viernes 10 de Abril.

1. **Dinámica de cazadores y de presas (20 puntos)**

Tenemos dos especies, una caza a la otra. El número de presas es x y el numero de cazadores es y . Su evolución temporal está descrita por las siguientes dos ecuaciones

$$\frac{dx(t)}{dt} = Ax(t) - Bx(t)y(t) \quad (1)$$

$$\frac{dy(t)}{dt} = -Cy(t) + Dx(t)y(t) \quad (2)$$

- Encuentre los valores de $x(0)$ y $y(0)$ para que exista equilibrio ($\dot{x} = \dot{y} = 0$) asumiendo que $A = 20$, $B = 1$, $C = 30$ y $D = 1$.
- Varíe las condiciones iniciales de la siguiente manera: $y(0)$ se mantiene fijo en el valor encontrado en el punto anterior mientras que $x(0)$ disminuye en una unidad con respecto al valor encontrado en el punto anterior hasta llegar a 1. Solucione el sistema de ecuaciones para $0 < t < 1$ y este conjunto de condiciones iniciales y represente las soluciones en el plano $x - y$.

El repositorio debe tener al menos tres códigos con los siguientes nombres: `volterra_lotka.c` que resuelve el sistema de ecuaciones diferenciales y `plot_solutions.py` que prepara los datos y la gráfica de las soluciones en el plano $x - y$.

Todo se debe poder ejecutar con un Makefile.

2. **Dinámica de tres masas** (30 puntos)

Escriba un programa que haga una integración con un Runge-Kutta de cuarto orden para el problema de las tres masas mencionado en clase para un tiempo de integración de 100 años. La única diferencia es que las velocidades iniciales van perpendiculares al plano definido por las tres masas.

3. **Colisiones de galaxias** (50 puntos)

Vamos a seguir el ejemplo del paper clásico de Toomre & Toomre *Galactic Bridges and Tails* en el *Astrophysical Journal*, Vol. 178, pp. 623-666 (1972), el cual se encuentra repositorio como `homework/2013-02/TT.pdf`

Una galaxia será descrita como un (1) cuerpo central de masa M con N cuerpos (estrellas) en órbitas que la rodean. Estos cuerpos van a estar distribuidos de manera uniforme sobre un círculo de radio R .

El movimiento **de cada una** de las partículas que rodean a la masa central está determinado por la siguiente ecuación diferencial ordinaria vectorial de segundo orden:

$$\frac{d^2 \vec{r}}{dt^2} = -\frac{GM}{r^2} \hat{r} \quad (3)$$

donde \vec{r} es un vector que va de la masa central a la partícula, \hat{r} es el vector unitario correspondiente, $r = |\vec{r}|$, M es la masa del cuerpo central y G es la constante de gravitación. Una vez se conocen las posiciones y velocidades iniciales de cada una de las masas \vec{r}_0 , \vec{v}_0 es posible conocer la posición y la velocidad en momentos siguientes. Noten que estamos usando una aproximación donde la fuerza que siente cada partícula solamente se debe solamente a la masa central. Esto es equivalente a decir a que la masa de todas las partículas es despreciable con respecto a la masa central. Así mismo la masa central no siente ninguna fuerza apreciable por parte de las partículas.

- (a) Escribir un programa que genere las condiciones iniciales (posiciones y velocidades) para que las N partículas de una galaxia de disco orbiten de manera estable en círculos alrededor de la masa central. El centro de la galaxia debe ubicarse en el punto $x_0 \hat{i} + y_0 \hat{j} + z_0 \hat{k}$ kpc y la velocidad del centro de masa es $v_{0x} \hat{i} + v_{0y} \hat{j} + v_{0z} \hat{k}$ km/s y donde $x_0, y_0, z_0, v_{0x}, v_{0y}$ y v_{0z} son números arbitrarios que se deben como argumento del código al momento de ser ejecutado. Igualmente los valores de M, R y N deben ser números que entran como parámetros de entrada al momento de la ejecución.

Para las cantidades de entrada las unidades son M_\odot , distancias en kpc y velocidades en km/s.

Estas condiciones iniciales (posiciones y velocidades) deben ser escritas en un archivo de texto al momento de ejecutar el código. En

este archivo la primera columna será un entero que llamaremos ID y corresponde a un número entero de cada partícula. El ID es negativo si se trata de la partícula central, y positivo si se trata de las partículas en las órbitas circulares.

- (b) Escribir el código que evolucione la posición y la velocidad de cada una de las partículas durante un tiempo T (medido en unidades de miles de millones de años). Este código debe leer las condiciones iniciales generadas por el código anterior. Se debe utilizar un método de Runge-Kutta de cuarto orden para integrar la ecuación de movimiento para cada partícula del disco. El código debe generar cinco archivos de texto con el entero de identidad, posiciones, velocidades de las partículas en M momentos diferentes equiespaciados en el tiempo T de evolución del sistema. El código debe tomar como argumento el nombre del archivo con las condiciones iniciales, el tiempo T y el número de momentos M . Cada vez que se escriba un archivo se debe usar el mismo formato utilizado para las condiciones iniciales.
- (c) Prepare un programa de python que toma como entrada un número arbitrario de archivos generados por el código anterior y prepara gráficas que muestran las posiciones. Utilice este programa para mostrar que las partículas que representan el disco de la galaxia siguen en su órbita circular después de 5 mil millones de años.
- (d) Este punto incluye una segunda galaxia idéntica a la primera. Si se considera que la galaxia anterior tiene un centro de masa en la posición $(0\hat{i} + 0\hat{j})$ kpc y una velocidad de centro de masa nula, vamos a considerar ahora que la masa central de la segunda galaxia tiene una posición $(150\hat{i} + 200\hat{j})$ kpc y el centro de masa tiene una velocidad inicial $-100\hat{j}$ km/s. La dirección \hat{z} es perpendicular al plano del disco.

Las condiciones iniciales de cada una de las galaxias son generadas con el código del primer punto con $N = 10^5$ y $R = 20$ kpc. Concatenando los dos archivos (utilizando el comando `cat`) debe generar un archivo único de condiciones iniciales.

Ahora evolucione estas nuevas condiciones iniciales por un intervalo $T = 5$ para ver la interacción de las dos galaxias. Prepare gráficas de la posición de las partículas de las dos galaxias en $M = 5$ momentos diferentes equiespaciados en el tiempo T .

Note que en esta configuración las masas centrales sienten su influencia mutua y su órbita también debe ser calculada.

El repositorio debe tener al menos tres códigos con los siguientes nombres:

- (a) **IC.c**: código que genera las condiciones iniciales de una sola galaxia con las posiciones y velocidades del centro de masa dadas como parámetros de entrada al momento de ejecutar el programa.

- (b) `evolve.c`: código que evoluciona un archivo con condiciones iniciales dadas al momento de ejecutar el programa.
- (c) `plots.py`: código que prepara gráficas de posiciones, tomando como entrada un número arbitrario de archivos de ID, posiciones y velocidades.

Todos los pasos deben poder ejecutarse con un Makefile. Inspiración: <http://viz.adrian.pw/galaxy/>

4. Colapso gravitacional (10/20/30 puntos)

Escriba un programa en C que integre las ecuaciones de movimiento de $N = 10^4$ masas iguales (cada una de $1M_{\odot}$) que se encuentran distribuidas inicialmente de manera aleatoria dentro de una esfera de radio 20 parsec. Modifique las ecuaciones de Newton para que la fuerza de gravedad sea una constante si la separación entre dos masas es menor que 0.01 parsec. Se debe describir la evolución del sistema por 10^9 años de tal manera que la energía total se conserve dentro de una incertidumbre menor a $\Delta E/|E| < 10^{-5}$.

Prepare gráficas de la evolución del sistema en 20 momentos diferentes, así como de la cantidad $G = 2K + U$ en cada paso de tiempo, donde K es la energía cinética y U es la energía potencial.

El repositorio debe tener al menos dos códigos. Uno llamado `nbody.c` que resuelve las ecuaciones diferenciales y otro `plot_nbody.py` que produce las gráficas necesarias.

Todo se debe poder ejecutar usando un Makefile.