Dinámica Molecular regida por el paso temporal

Trabajo Práctico Nro. 4

Badi Leonel, Buchhalter Nicolás Demián y Meola Franco Román

2 de mayo de 2016

Grupo 3

Que vamos a ver

Oscilador Puntual Amortiguado
Formación del Sistema Solar

Parte I

Oscilador Puntual Amortiguado

Fundamentos
Introducción

- Vamos a comparar los errores cometidos por distintos sistemas de integración
- Oscilador amortiguado: Sistema con sólo una partícula puntual cuya solución analítica es conocida
- Se implementaron:
    - *Beeman*
    - *Velocity Verlet*
    - *Gear Predictor Corrector de orden 5*

Fundamentos

Variables relevantes

- **Parámetros del oscilador**
  - $m = 70$
  - $k = 10000$
  - $\gamma = 100$
  - $t_f = 5$
- **Condiciones iniciales del oscilador**
  - $r_0 = 1$
  - $v_0 = -\frac{2\gamma}{m}$

Implementación
Cálculo Numérico

```
void simulateGear(double time, double deltaT) {
    double simTime = 0;
    Oscilator oscilator = new Oscilator();
    oscilator.writePositionAndError();
    oscilator.makeEulerStep(deltaT);
    simTime += deltaT;
    oscilator.writePositionAndError();
    while (simTime < time) {
        oscilator.makeGearStep(deltaT);
        simTime += deltaT;
        oscilator.writePositionAndError();
    }
}
```

Código 1: Método de Gear Predictor Corrector

Implementación

Detalles de precisión

- Todas las operaciones se realizan en `double`
- Se utilizan cinco cifras decimales como `output` en los archivos de salida de resultados y errores

Resultados

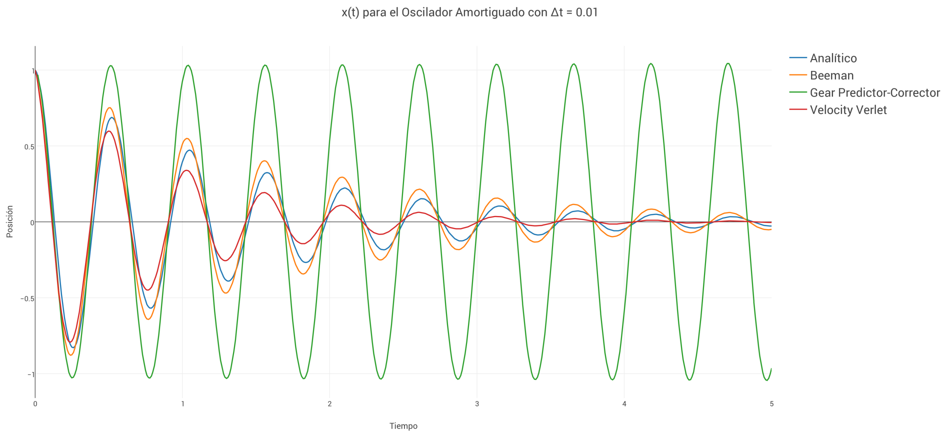
Error total normalizado por el número total de pasos para distintos valores de Δt

Δt	**Método**	E
0.01	*Beeman*	0,00471
0.01	*Verlet*	0,00663
0.01	*Gear*	0,33624
0.001	*Beeman*	0,00235
0.001	*Verlet*	0,00225
0.001	**Gear**	**-0,00199**
0.0001	*Beeman*	0,00225
0.0001	*Verlet*	0,00224
0.0001	*Gear*	0,00228

Tabla: Suma de las diferencias al cuadrado para todos los pasos temporales normalizado por el número total de pasos

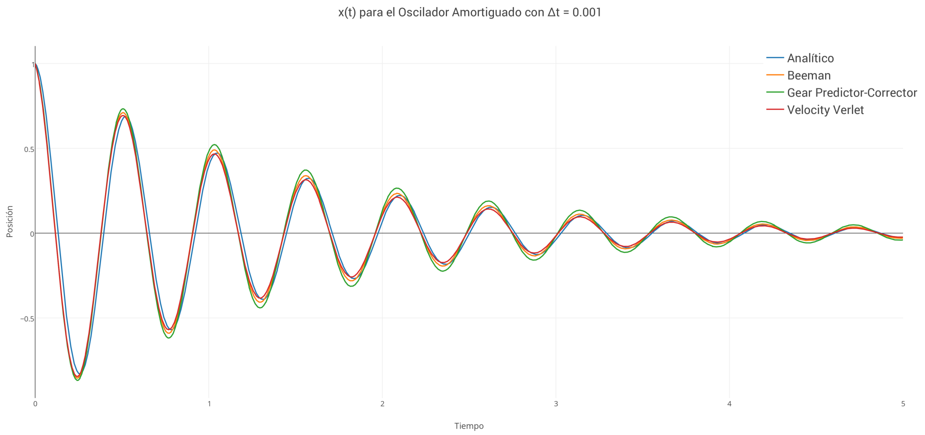
Resultados

Gráfico de $x(t)$ para el oscilador puntual amortiguado con $\Delta t = 0{,}01$



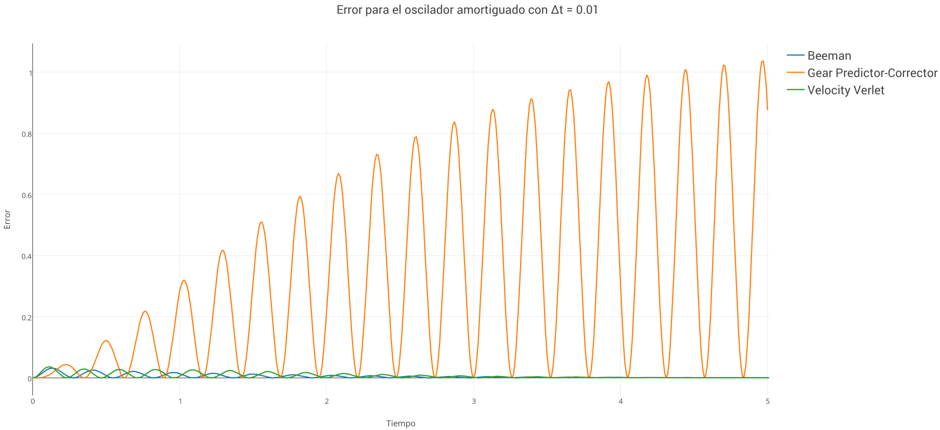
Resultados

Gráfico de $x(t)$ para el oscilador puntual amortiguado con $\Delta t = 0{,}001$



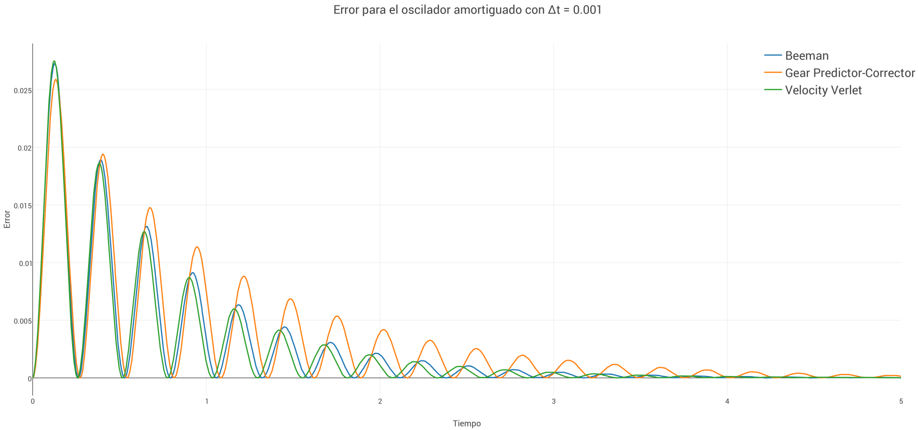
Resultados

Gráfico de E para el oscilador puntual amortiguado con $\Delta t = 0{,}01$



Resultados

Gráfico de E para el oscilador puntual amortiguado con $\Delta t = 0{,}001$



Conclusiones

- Para una cantidad de pasos baja (500 pasos, $\Delta t = 0{,}01$), el error de *Gear Predictor-Corrector* aumenta, simulando un oscilador no amortiguado
- Con un $\Delta t = 0{,}001$ obtuvimos resultados con errores muy bajos para los tres métodos
- Con 50000 pasos ($\Delta t = 0{,}0001$), los tres métodos tienen un error que varía recíen en la quinta cifra decimal
- El esquema de integración que mejor resulta para este sistema es *Gear Predictor-Corrector* para $\Delta t = 0{,}001$, es decir, 5000 pasos

Parte II

Formación del Sistema Solar

Fundamentos
Introducción

- Usando el esquema de integración de *Beeman* vamos a simular el nacimiento del sistema solar
- Se simularán N partículas que orbitan alrededor del Sol
- Las partículas se irán agrupando en planetas a medida que el sistema evolucione

Implementación

Generación de los agentes

- Posiciones (x, y) aleatorias para todas las partículas
- v_{t_0} tal que todas las partículas tengan el mismo L
- $v_{n_0} = 0$
- Distancia al sol entre 1×10^9 y 1×10^{10}
- Ángulo respecto al Sol ϵ $[0, 2\pi]$

Simulación

Variables relevantes

- Δt: cantidad de pasos
- k: relación entre cantidad de pasos simulados y visualizados.
- `time`: Tiempo en segundos a visualizar

Simulación

Detalles de implementación

- Utilizamos el *Cell Index Method* para calcular las colisiones de las partículas
- Para las partículas que se alejen más de 2×10^4 del centro, no las consideramos dentro del sistema
- Para simplificar, luego de la colisión de dos partículas, se obtiene una nueva con un radio correspondiente a la suma de los radios de las dos

Simulación

Problemas encontrados

- **Tratamiento de números de grandes dimensiones**
    - Necesitamos poder mantener en memoria números grandes utilizando la precisión `double`
    - Se normalizó r a 1×10^6
    - Se normalizó m a 2×10^{25}
- **El radio de las partículas es muy chico en comparación con las dimensiones del sistema solar**
    - Esto dificulta la visualización, sobre todo para una gran cantidad de partículas
    - El r_c es distinto al r_v (radio de visualización)

Implementación

Simulación

```
void simulate(int k, double dt, int time){
    write();
    moveEuler(dt);
    int framesWrited = 1;
    double totalTimeSimulated = dt;
    while(totalTimeSimulated < time) {
        for(int i=0; i<k; i++) {
            moveBeeman(dt);
            findNeighbours();
            collidePlanets();
            totalTimeSimulated += dt;
            write();
        }
        write();
        framesWrited++;
    }
}
```

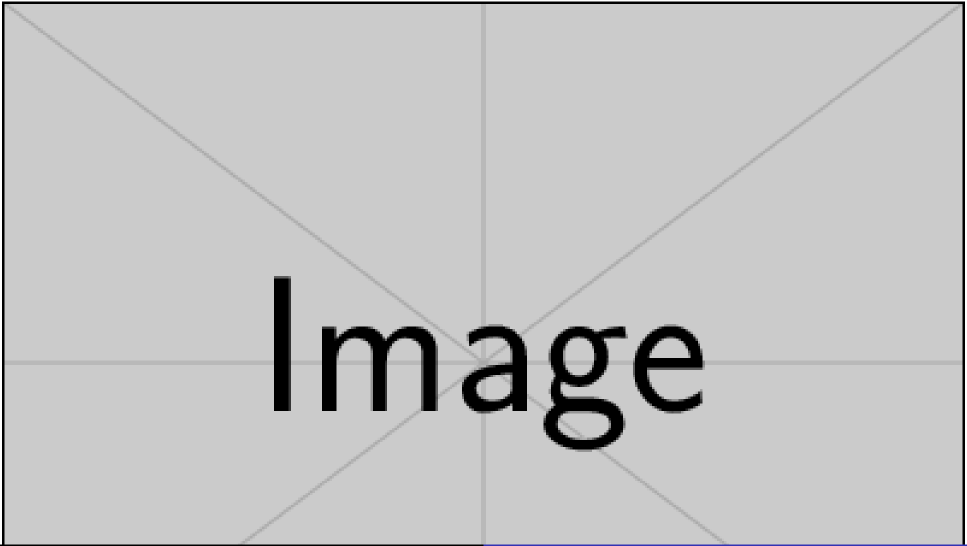
Implementación

Visualización

- La simulación y la visualización son independientes
- El algoritmo de simulación escribe un archivo `.tsv` con los siguientes datos:
  - (x, y)
  - r
  - Color RGB para indicar las velocidades, donde R es la componente en el eje Y y G es la componente en eje X
- Por último, se carga en `Ovito` el archivo de salida `.tsv` para realizar la visualización

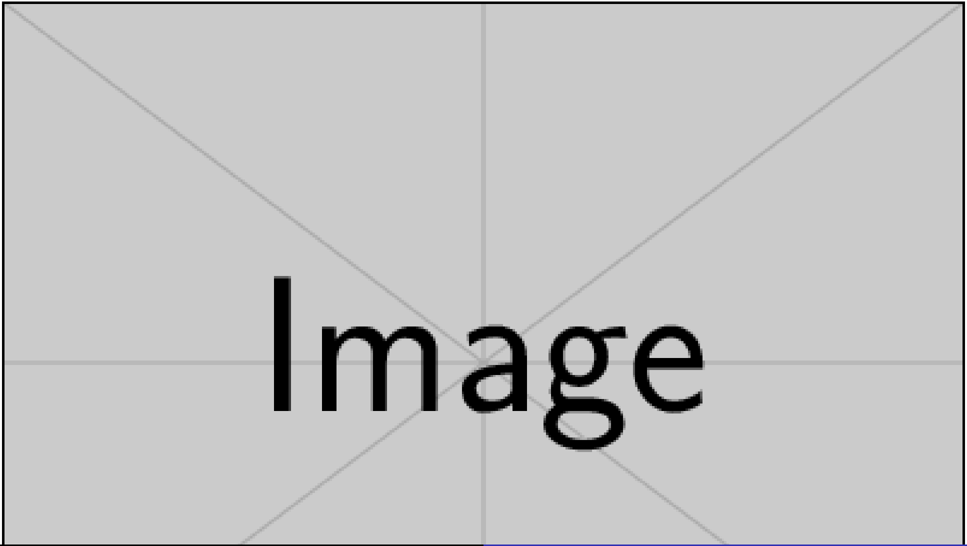
Resultados

Gráfico de las energías U, K y E_T para la simulación de $N = 100$



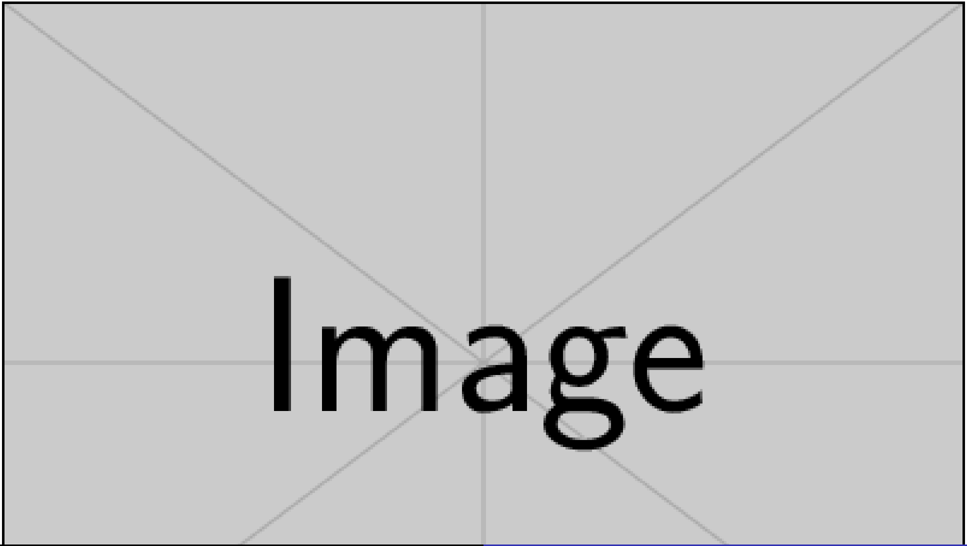
Resultados

Gráfico de las energías U, K y E_T para la simulación de $N = 1000$



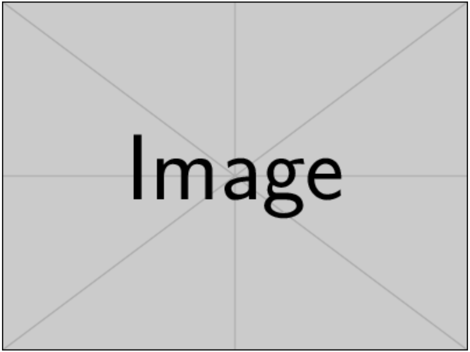
Resultados

Gráfico de las energías U, K y E_T para la simulación de $N = 10000$



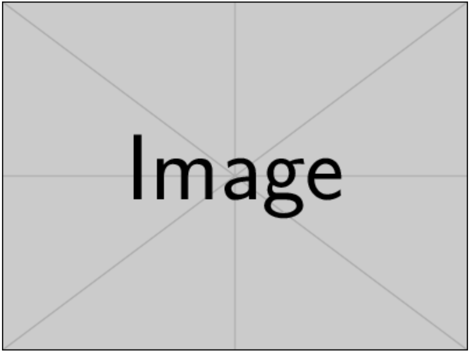
Resultados

Animación de la simulación para $N = 100$



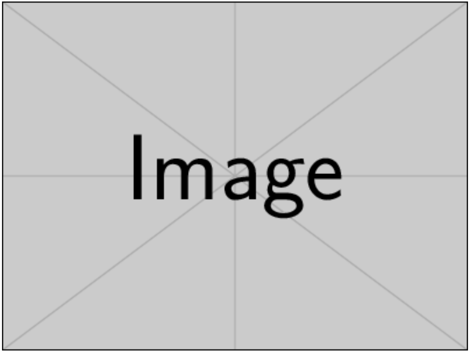
Resultados

Animación de la simulación para $N = 1000$



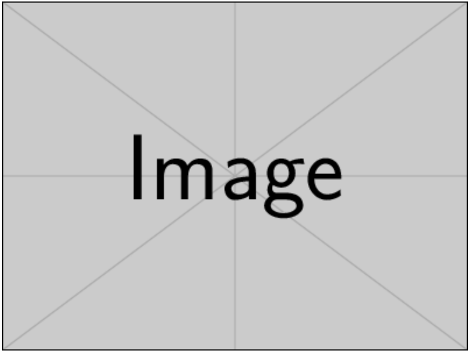
Resultados

Animación de la simulación para $N = 10000$



Resultados

Animación de la simulación para $N = 50000$



Conclusiones

- El paso temporal (Δt) óptimo para simular el sistema es ?.

Gracias