



*Universidad Michoacana de San Nicolás de Hidalgo*

*Instituto de Física y Matemáticas*

# Optimización de datos iniciales para el método SPH

Tesis

que para obtener el grado de

**Maestro en Ciencias en el área de Física**

presenta

**Silvano Ulices Que Salinas**

Morelia, Michoacán. Febrero del 2014

Director de tesis:  
Doctor en ciencias en el área de física José Antonio Gonzáles Cervera

Instituto de Física y Matemáticas,  
UMSNH.

## *Dedicatoria*

La aventura más grande y persistente de la historia humana es la búsqueda de la comprensión del Universo, de su funcionamiento y su origen. Resulta difícil pensar que un puñado de seres, habitantes de un minúsculo planeta que gira en torno a una estrella insignificante perteneciente a una pequeña galaxia, se hayan propuesto el objetivo de alcanzar la comprensión total del cosmos, resulta difícil pensar que una partícula ínfima de la creación confíe plenamente en su capacidad de entender la totalidad.

Murria Gell-Mann

## *Agradecimientos*

x

x

x

x



# Índice general

Contenido	I
Lista de figuras	VII
Lista de tablas	XI
Abstract	1
Resumen	1
Motivación.	2
Antecedentes.	5
<b>1. Optimización numérica multivariable sin restricciones.</b>	<b>15</b>
1.1. Métodos de relajación. . . . .	15
1.2. Método del gradiente. . . . .	16
1.3. Búsqueda lineal inexacta del paso. . . . .	17
1.4. Método del gradiente conjugado. . . . .	19
<b>2. Construcción del proceso de optimización.</b>	<b>21</b>
2.1. Resultados iniciales. . . . .	21
2.2. Construcción del método. . . . .	25
2.3. Análisis preliminar del método. . . . .	28
2.4. Parámetros iniciales. . . . .	34

2.5.	El problema de los pares. . . . .	37
<b>3.</b>	<b>Análisis de resultados.</b>	<b>41</b>
3.1.	Caso perfil de densidad constante en 2D. . . . .	41
3.1.1.	Método del gradiente. . . . .	41
3.1.2.	Método del gradiente conjugado. . . . .	46
3.1.3.	Búsqueda lineal inexacta del paso. . . . .	48
3.2.	Caso perfil de densidad constante en 3D. . . . .	50
3.2.1.	Método del gradiente. . . . .	50
3.2.2.	Método del gradiente conjugado. . . . .	52
3.3.	Caso perfil de densidad dependiente de r en 3D. . . . .	54
3.3.1.	Fronteras aleatorias fijas. . . . .	55
3.3.2.	Método del gradiente. . . . .	57
3.3.3.	Método del gradiente conjugado. . . . .	58
<b>4.</b>	<b>Conclusiones.</b>	<b>61</b>
4.1.	Perspectivas. . . . .	64
<b>A.</b>	<b>Método de las direcciones conjugadas.</b>	<b>65</b>
<b>Apéndices</b>		<b>65</b>
<b>B.</b>	<b>Buscador eficiente de vecinos</b>	<b>69</b>
B.1.	Clasificación de las partículas . . . . .	70
B.2.	Ordenamiento de raíz. . . . .	71
B.3.	Censo de las partículas. . . . .	73
B.4.	Búsqueda de las partículas. . . . .	74
B.5.	Tiempo de procesamiento. . . . .	75
<b>C.</b>	<b>Anticolisionador.</b>	<b>77</b>
<b>Bibliografía</b>		<b>81</b>

# Índice de figuras

1.	Ejemplo del método de aceptación y rechazo para una función de distribución de probabilidad acotada. . . . .	8
1.1.	Criterio de Armijo; la sección de la curva $f(\mathbf{x} + \alpha\mathbf{S})$ , que se encuentra por debajo de la línea punteada que representa a $f(\mathbf{x}) + \rho\alpha\nabla f(\mathbf{x} + \alpha\mathbf{S})^T\mathbf{S}$ (concretamente desde $\alpha = 0$ hasta $\alpha = \alpha_0$ ), corresponde al conjunto de tamaños de paso que satisfacen la desigualdad. . . . .	18
2.1.	Posiciones de las partículas generadas mediante el método de aceptación y rechazo (en coordenadas cartesianas) para el sistema descrito en la ecuación 2.1. . . . .	23
2.2.	Comparativa del perfil de densidad a la SPH para las posiciones iniciales de los puntos generados mediante el método de aceptación y rechazo con respecto al perfil de densidad analítico en el caso del sistema asociado con la ecuación 2.1. . . . .	24
2.3.	Configuración de datos iniciales caracterizada por 1296 partículas ordenadas en una cuadrícula de 36x36 posiciones. . . . .	29
2.4.	Perfil de densidad calculado a la SPH de una configuración ordenada de datos iniciales con densidad analítica constante igual a 1. Los puntos en las fronteras (que en la gráfica son los puntos por debajo del valor deseado) tienen un valor que decae debido a la falta de vecinos en la frontera que les permita calcular un buen promedio. . . . .	30
2.5.	Perfil del promedio de la derivada de la densidad calculado a la SPH de una configuración ordenada de datos iniciales con densidad analítica constante igual a 1. El comportamiento en las fronteras se debe a la caída del perfil de densidad en la misma zona. . . . .	30

2.6.	Configuración inicial de 576 partículas aleatorias a relajar generadas mediante el método de aceptación y rechazo con 720 partículas como frontera fijas. El proceso de relajación mediante el método del gradiente se ejecutará sólo sobre las partículas aleatorias. . . . .	32
2.7.	Tendencia de la función error durante el proceso de relajación para una configuración de 576 partículas a relajar. . . . .	34
2.8.	Tendencia de la función error durante el proceso de relajación en escala logarítmica para un configuración de 4 partículas a relajar. La curva muestra el rápido descenso en la magnitud del desplazamiento promedio de las partículas. La rápida caída en la curva se debe puramente al método del gradiente aplicado al método SPH; este patrón se verá repetidamente mostrado en los resultados presentados a lo largo de este trabajo. . . . .	36
3.1.	Gráfica de la magnitud promedio del desplazamiento de las partículas contra el tiempo durante el proceso de relajación de una configuración constante en 2D de 10000 partículas aleatorias. . . . .	43
3.2.	Análisis del perfil de densidad final e inicial para una configuración con densidad constante en 2D, para 576 partículas aleatorias con una longitud $h$ fija. . . . .	49
3.3.	Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 3D para 13824 partículas aleatorias con una longitud $h$ fija. . . . .	51
3.4.	Gráfica en escala logarítmica del desplazamiento promedio de las partículas durante el proceso de relajación. . . . .	52
3.5.	Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 3D para 13824 partículas aleatorias con una longitud $h$ fija. . . . .	53
3.6.	Análisis del perfil de densidad final e inicial para un configuración dependiente del radio en 3D de 3089 partículas aleatorias, con una longitud $h$ fija y fronteras conformadas por partículas aleatorias. . . .	58
3.7.	Análisis del perfil de densidad final e inicial para un configuración dependiente del radio en 3D de 3089 partículas aleatorias, con una longitud $h$ fija. . . . .	59
4.1.	Gráfica comparativa del error absoluto para los métodos del gradiente, búsqueda inexacta del paso y del gradiente conjugado para un configuración de 4096 partículas en 2D. . . . .	62

4.2. Gráfica comparativa del error absoluto para los métodos del gradiente y del gradiente conjugado para un configuración de 4096 partículas en 3D. . . . .	63
4.3. Gráfica comparativa del error absoluto para los métodos del gradiente y del gradiente conjugado para un configuración de 4096 partículas en 3D. . . . .	63
A.1. Direcciones Conjugadas. . . . .	66
A.2. Método de Powell. . . . .	67
B.1. Comparativa del tiempo de procesamiento entre la versión del código con buscador de vecinos y la versión sin buscador. . . . .	75
C.1. Comparativa del tiempo de procesamiento entre la versión del código con Anticolisionador y la versión sin él. . . . .	79



# Índice de cuadros

2.1. Proceso de relajación para 576 partículas ordenadas y compactadas en el centro de la cuadrícula. Con una frontera de 720 partículas ordenadas fijas. De izquierda a derecha y de arriba hacia abajo se observa como las partículas aleatorias van modificando su posición al someterse al proceso de optimización sin que se observe una dirección preferente de desplazamiento. . . . .	33
2.2. Proceso de relajación para 4 partículas aleatorias ubicadas en el centro de la cuadrícula. La figura a la izquierda muestra la posición de las mismas antes de aplicar el proceso de relajación y la figura a la derecha muestra a la optimización de las posiciones. Las partículas al tener muchos vecinos ya ordenados y fijos, obtienen un buen promedio para su densidad y para la derivada de la misma. . . . .	35
2.3. Proceso de relajación para partículas aleatorias centrales rodeadas por una amplia frontera de partículas fijas. A la izquierda se muestran las configuraciones iniciales de 16 (panel superior) y 36 (panel inferior) partículas aleatorias; a la derecha se muestra el resultado de aplicar el proceso de relajación. Tras el proceso de relajación observan partículas que se juntan formando pares afectando los promedios calculados a la SPH. . . . .	38
2.4. Análisis del desplazamiento de las partículas (en escala logarítmica en el panel izquierdo) y su vector de desplazamiento (panel derecho). Aunque las partículas se siguen desplazando hacia una posición óptima, la magnitud del desplazamiento promedio de las partículas decrece rápidamente. . . . .	39
3.1. Proceso de relajación para una configuración con densidad constante de 4096 (panel superior) y 10000 (panel inferior) partículas aleatorias en 2D mediante el método del gradiente. . . . .	42

3.2.	Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 4096 (panel izquierdo) y 10000 (panel derecho) partículas a relajar empleando una longitud $h$ fija. . .	44
3.3.	Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 576 (panel izquierdo) y 4096 (panel derecho) partículas aleatorias empleando una longitud $h$ variable. . .	44
3.4.	Comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes $h$ fija y $h$ variable, en una configuración con densidad constante 2D de 4096 partículas aleatorias.	45
3.5.	Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos en los que se emplean longitudes $h$ fija y $h$ variable, en una configuración con densidad constante 2D de 4096 partículas a relajar; las gráficas inferiores representan los casos respectivos del error absoluto (a la izquierda) y la función error (a la derecha) con el uso del anticolidionador. . . . .	46
3.6.	Proceso de relajación para una configuración con densidad constante de 576 (panel superior) y 4096 (panel inferior) partículas aleatorias en 2D, mediante el método del gradiente conjugado. . . . .	47
3.7.	Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 576 (panel izquierdo) y 4096 (panel derecho) partículas aleatorias con una longitud $h$ fija. . . . .	47
3.8.	Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes $h$ fija, $h$ variable y $h$ fija con anticolidionador, para una configuración con densidad constante 2D de 4096 partículas aleatorias. . . . .	48
3.9.	Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con una longitud $h$ fija y $h$ variable, para una configuración con densidad constante 2D de 576 partículas aleatorias. . . . .	49
3.10.	Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con una longitud $h$ fija sin anticolidionador y una $h$ fija con anticolidionador para una configuración con densidad constante 3D de 4096 partículas aleatorias. . . .	52
3.11.	Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes $h$ fija sin anticolidionador y $h$ fija con anticolidionador para una configuración con densidad constante 3D de 4096 partículas aleatorias. . . . .	54

3.12. Análisis del proceso de optimización empleando fronteras aleatorias para un caso con 4096 partículas a relajar para una densidad constante. Se observa que el perfil de densidad (panel izquierdo) obtiene una mejora considerable con sólo 10000 iteraciones. La gráfica del error absoluto (panel derecho) refleja la tendencia en la reducción del mismo. . . . .	57
3.13. Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes $h$ fija sin anticolidionador, $h$ fija con anticolidionador y $h$ variable para una configuración dependiente del radio en 3D de 3089 partículas aleatorias, empleando el método del gradiente conjugado. . . . .	58
3.14. Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes $h$ fija sin anticolidionador, $h$ fija con anticolidionador y $h$ variable para una configuración dependiente del radio en 3D de 3089 partículas aleatorias, empleando el método del gradiente conjugado. . . . .	59

# Abstract

In this paper an algorithm has been developed aimed at optimizing the positions of the particles SPH (smooth particle hydrodynamics) method uses as initial data representing a density profile characteristic of a certain physical phenomenon to be studied. To this end, we conducted a survey of various optimization methods that use the gradient as a tool to reach the minimum of a certain function.

**Keywords:** SPH, optimization, gradient method, acceptance-rejection method.

# Resumen

En este trabajo se ha elaborado un algoritmo enfocado a optimizar las posiciones de partículas que el método SPH (*smooth particle hydrodynamics*) emplea como datos iniciales y que representan a un perfil de densidad característico de cierto fenómeno físico que se desea estudiar. Con este fin, se lleva a cabo un estudio de diversos métodos de optimización que emplean al gradiente como herramienta para llegar al mínimo de una cierta función.

**Palabras clave:** SPH, optimización, método del Gradiente, método de aceptación y rechazo.

# Motivación

## Generación e importancia de los datos iniciales.

Imaginemos que deseamos modelar un proceso astrofísico tal como el choque de un agujero negro y un cúmulo de polvo cósmico, una estrella o una galaxia. Sabemos que para ello necesitaremos hacer uso de la relatividad general; así como de la dinámica de fluidos, debido al carácter físico intrínseco en dicho sistema.

El deseo de construir modelos de tales sistemas radica en la necesidad de describir los fenómenos mismos con la mayor aproximación posible, construyendo con ello predicciones numéricas asociadas al fenómeno mismo. Para poder llevar a cabo esto, es necesario resolver de forma discreta las ecuaciones físicas que describen al sistema y llevar a cabo de algún modo su evolución en el tiempo.

Una forma de llevar a cabo esto es justamente mediante el método de SPH (*Smooth Particle Hydrodynamics*) [1][2][3], el cual tiene una especial aplicación en problemas de carácter astrofísico, que fue el área en la originalmente fue desarrollado; en la actualidad el método SPH tiene una importante aplicación en otras áreas tales como balística, vulcanología y oceanografía; incluso SPH es usado en la industria del entretenimiento para la elaboración de animaciones en tiempo real o para videojuegos. SPH es un método empleado para discretizar ecuaciones, que permite obtener una aproximación numérica a las soluciones de las ecuaciones de la dinámica de fluidos reemplazando al fluido mismo por un conjunto de partículas.

Así, dichas partículas pueden ser vistas como la interpolación de las posiciones de los elementos del fluido, desde las cuales las propiedades físicas del fluido pueden ser calculadas. El proceso para obtener datos iniciales<sup>1</sup> en SPH es por esto de carácter fundamental, debido a que la eficiencia de todo el proceso evolutivo tendrá una

---

<sup>1</sup>Con “datos iniciales”, nos estaremos refiriendo a lo largo de este trabajo a las posiciones de los elementos del fluido (que comúnmente llamaremos “partículas” por simplicidad) que se emplean como parte de las condiciones iniciales, las cuales emplea el método de SPH para llevar a cabo algún proceso determinado.

correlación con la eficiencia en la obtención de datos iniciales. Esto dado que el proceso que se desarrolle mediante el método SPH nos devolverá resultados que dependen con relativa importancia de la posición de cada una de las partículas que conforman al sistema, lo cual quiere decir que a un tiempo  $t = 0$  necesitamos un perfil de densidad conformado por aquellas partículas, que represente lo más fielmente posible a nuestro fenómeno físico a fin de que el proceso evolutivo que se lleva a cabo pueda tener un significado físico real.

## El problema con el perfil de densidad.

La generación de datos iniciales para SPH se lleva a cabo mediante el método de aceptación y rechazo que es parte de los métodos de simulación tipo Monte Carlo. Este método nos permite generar los datos iniciales de forma simple, rellenando el volumen  $3N$ -dimensional<sup>2</sup> con partículas, tal como si las arrojáramos dentro del mismo de forma aleatoria.

El primer problema que se presenta al hacer esto, es que no sabemos si el perfil de densidad de los datos iniciales generados por dicho proceso, se aproxima lo suficiente al perfil de densidad analítico que representa al sistema físico que se desea modelar mediante el método SPH. Por ello se debe llevar a cabo un estudio de los datos iniciales a fin de encontrar cuál es su respectivo perfil de densidad y comparar a éste con el perfil de densidad analítico deseado<sup>3</sup>.

Esto nos debería arrojar una nube de puntos que deberá acercarse de buena manera al perfil analítico deseado, pero sucede que al comparar las posiciones de dichos puntos (que representan a los elementos del fluido o partículas), se observa una clara diferencia con respecto a las posiciones de los puntos que conforman al perfil de densidad analítico.

Es fundamental que los datos iniciales tengan un perfil lo mas parecido posible al perfil inicial deseado, puesto que la fidelidad de los valores numéricos obtenidos que representarán la física de nuestro sistema dependerá de la precisión que tenga el proceso evolutivo mediante SPH, mismo que, a su vez depende justamente del perfil de densidad inicial.

Tenemos, por tanto, el problema de cómo minimizar la dispersión<sup>4</sup> del perfil obtenido para los datos iniciales, a fin de que las posiciones de los puntos que con-

---

<sup>2</sup> $N$  es el número de partículas que conforman al sistema físico.

<sup>3</sup>En este trabajo no se ha realizado la modelación mediante el método SPH de fenómeno físico alguno. El objetivo del mismo solo está en generar datos iniciales que representen a un perfil de densidad lo más aproximado posible a la realidad con respecto a un cierto fenómeno físico de interés.

<sup>4</sup>Dispersión de las posiciones para los puntos generados mediante el método de aceptación y

forman al perfil inicial generado estén prácticamente en la misma posición que los puntos que conforman al perfil analítico deseado.

---

rechazo con respecto a las posiciones de los puntos que conforman el perfil deseado.

---

# Antecedentes

## Método de Monte Carlo.

El método de Monte Carlo es un método numérico que permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias. La importancia del método de Monte Carlo se reconoce cuando se aplica a problemas que tienen difícil solución por métodos exclusivamente analíticos o numéricos (resolución de integrales de muchas variables, minimización de funciones, etc.). Algunos de estos problemas dependen de factores aleatorios o se pueden asociar a modelos probabilísticos.

Gracias al avance en el diseño de los ordenadores, cálculos tipo Monte Carlo, que en otro tiempo hubieran sido inconcebibles, hoy en día se presentan como alternativas para la resolución de ciertos problemas. En estos métodos el error es aproximadamente igual a  $1/\sqrt{N}$ [4], donde  $N$  es el número de pruebas y, por tanto, ganar una cifra decimal en la precisión implica aumentar  $N$  en 100 veces. La base del método está en la generación de números aleatorios de los que nos serviremos para calcular probabilidades. Conseguir un buen generador de números aleatorios, así como un conjunto estadístico adecuado sobre el que trabajar, son las primeras dificultades con la que nos vamos a encontrar al momento de utilizar este método.

El método de Monte Carlo representa a una amplia clase de algoritmos computacionales que emplean un repetitivo muestreo aleatorio para obtener resultados numéricos; es decir, mediante la ejecución repetitiva de muchas simulaciones. Éstas se realizan con el fin de calcular las mismas probabilidades heurísticamente tal como si estuviésemos jugando en un casino real y anotando nuestros resultados.

El método de Monte Carlo se utiliza principalmente en tres problemas diferentes: optimización, integración numérica y generación de muestras a partir de una distribución de probabilidad. Este método es especialmente útil para la simulación de sistemas con muchos grados de libertad acoplados, tales como los fluidos, entre muchos otros.

---

No hay consenso sobre cómo debe definirse el método de Monte Carlo. Por ejemplo, Sawilowsky [5] distingue entre una simulación, un método de Monte Carlo, y una simulación de Monte Carlo: una simulación es una representación ficticia de la realidad, un método de Monte Carlo es una técnica que se puede utilizar para resolver un problema matemático o estadístico, y una simulación de Monte Carlo son los usos repetidos de toma de muestras para determinar las propiedades de algún fenómeno (o comportamiento). Lo anterior se puede entender con los siguientes ejemplos:

-Simulación: Una variable uniforme pseudo-aleatoria definida desde el intervalo  $[0, 1]$  se puede utilizar para simular el lanzamiento de una moneda: si el valor es menor o igual a 0.50, se designa el resultado como cara, pero si el valor es mayor a 0.50, se designa el resultado como cruz. Esto es una simulación, pero no una simulación de Monte Carlo.

-Método de Monte Carlo: El área de una figura irregular inscrita en un cuadrado unitario puede ser determinada por el número total de dardos lanzados dentro del área. Este es un método de Monte Carlo para determinar el área, pero no una simulación.

-Simulación de Monte Carlo: Definiendo un gran número de variables uniformes pseudo-aleatorias en el intervalo  $(0, 1]$ , y designando a valores menores o iguales a 0.50 como caras, y mayores que 0.50 como cruz, tendremos una simulación de Monte Carlo para el lanzamiento repetitivo de una moneda.

El método de simulación de Monte Carlo no siempre requiere números verdaderamente aleatorios para ser útil. Mientras que para algunas aplicaciones la impredecibilidad es vital, muchas de las técnicas más útiles aplicadas a la programación numérica usan secuencias deterministas pseudo-aleatorias, por lo que es fácil de probar y volver a ejecutar las simulaciones. La única cualidad necesaria para tomar buenas simulaciones suele ser que la ‘secuencia pseudo-aleatoria’ parezca ser lo “bastante aleatoria”.

## Generación de números pseudo-aleatorios.

Se llama números pseudo-aleatorios  $\{u_i\}_{i \geq 1}$  a una sucesión determinística de números en el intervalo  $[0, 1]$ ; que tiene las mismas propiedades estadísticas que una sucesión de números aleatorios. Una forma general de obtener números pseudo-aleatorios es partir de una semilla de  $p$  números  $u_{-p+1}, u_{-p+2}, \dots, u_{-1}, u_0$  y aplicar una función  $d$  de modo que  $u_i = d(u_{i-1}, \dots, u_{i-p})$ .

Un generador pseudo-aleatorio de números es un algoritmo que produce una sucesión de números que es una muy buena aproximación a un conjunto aleatorio

---

de números. La sucesión no es exactamente aleatoria, en el sentido de que queda completamente determinada por un conjunto relativamente pequeño de valores iniciales.

La mayoría de los algoritmos de generadores pseudo-aleatorios producen sucesiones que poseen una serie de propiedades que son las que les permiten a estas ser usadas en lugar de números realmente aleatorios. Las propiedades mínimas que deben satisfacer los números pseudo-aleatorios, en general, son:

- Ajustarse a una distribución  $U(0, 1)$  (Distribución uniforme continua con rango entre 0 y 1).
- Ser estadísticamente independientes (no debe deducirse un número conociendo otros ya generados).
- Ser reproducibles (la misma semilla debe dar la misma sucesión).
- Tener un Ciclo repetitivo muy largo<sup>5</sup>.
- Facilidad de obtención.
- Ocupar poca memoria.

## Método de aceptación y rechazo.

El método de aceptación y rechazo es un método de Monte Carlo, el cual emplea secuencias de números pseudo-aleatorios con el fin de representar a variables aleatorias a partir de las cuales se lleva a cabo una simulación.

Este método básicamente consiste en muestrear una variable aleatoria respecto a una función de distribución apropiada, y someter a dicha variable, a un análisis para determinar si se acepta o se rechaza. Técnicamente, lo que se hace es expresar de forma adecuada a la función de distribución respecto a la cual se quiere obtener valores de una variable aleatoria, para entonces imponer un criterio mediante el cual estos valores serán aceptados.

Una variable aleatoria o variable estocástica, es una variable cuyos valores se obtienen de mediciones de algún tipo de experimento de carácter aleatorio. Formalmente, una variable aleatoria es una función, la cual asigna eventos a números reales. Puede concebirse también como un valor numérico que está afectado por el azar.

Sea  $x$  la variable aleatoria de la cual se quiere obtener valores distribuidos con la función de distribución de probabilidad  $F(x)$  con  $x \in \mathbb{I}$ , donde  $\mathbb{I}$  representa a los números irracionales.

---

<sup>5</sup>Esto dado que ciclos cortos limitan la longitud aprovechable de una corrida de una simulación dado que el reciclaje resulta en una repetición de secuencias de eventos.

---

Entonces se generan dos variables aleatorias  $a$  y  $b$ , distribuidas uniformemente en  $(0, 1)$ ; se reescalan ambas variables de la forma apropiada a fin de obtener tanto  $x$  como  $y$ , y se comprueba si se cumple la desigualdad  $y \leq F(x)$ . Si se cumple, se acepta  $x$  como valor de una variable aleatoria con función de densidad de probabilidad  $F(x)$ . En caso contrario, se rechazan los valores de  $a$  y  $b$ , generando dos nuevos valores para estas variables.

Un ejemplo del resultado del proceso de emplear el método de aceptación y rechazo es generar variables aleatorias que obedezcan a una función de probabilidad para un caso particular en dos dimensiones tal como se muestra en la figura 1. La curva allí mostrada representa a una cierta función de densidad de probabilidad  $F(x)$ , la cual tiene un valor máximo  $M$ .

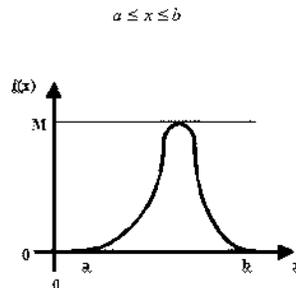


Figura 1: Ejemplo del método de aceptación y rechazo para una función de distribución de probabilidad acotada.

Entonces generamos dos números aleatorios  $U_1$  y  $U_2$ , y determinamos el valor de la variable aleatoria  $x$  de acuerdo a la siguiente relación lineal de  $U_1$ :

$$x = a + (b - a)U_1. \quad (1)$$

Evaluamos la función de probabilidad en  $x = a + (b - a)U_1$  y analizamos si la siguiente desigualdad se cumple:

$$U_2 \leq \frac{f(a + (b - a)U_1)}{M}. \quad (2)$$

Se utilizará a  $x = a + (b - a)U_1$  si la respuesta es afirmativa, como un valor simulado de la variable aleatoria que se desea generar. De lo contrario, es necesario generar nuevamente nuevos números aleatorios tantas veces como sea necesario.

---

Este método es sencillo de aplicar siempre y cuando la función de distribución no tenga una fórmula complicada y utilice sólo un número aleatorio para calcular un valor de la variable aleatoria.

## Nociones básicas de SPH.

Tal como se había comentado con anterioridad, SPH es un método para obtener aproximaciones numéricas a las soluciones de las ecuaciones de la dinámica de fluidos mediante el reemplazamiento de los fluidos por un conjunto de partículas.

Para los matemáticos, dichas partículas son justamente la interpolación de las posiciones de los elementos del fluido, a partir de las cuales las propiedades de los fluidos pueden ser calculadas. Para los físicos, las partículas en SPH son partículas materiales, las cuales pueden ser tratadas como cualquier otro sistema de partículas.

Una de las ventajas del uso del método SPH es que se puede usar con más de un tipo de material. En este caso cada material puede ser descrito por un conjunto propio de partículas; con ello, problemas de interfase son a menudo triviales de describir en SPH, mientras que pueden ser difíciles en esquemas de diferencias finitas. Otra ventaja es que el método proporciona un puente natural entre la versión discreta y continua del sistema.

Habría que considerar también la ventaja de que la resolución puede hacerse dependiente de la posición y el tiempo, lo cual hace al método muy atractivo para la mayoría de los problemas astrofísicos y geofísicos [3][6][7]. Finalmente, dada la cercana similitud entre SPH y la dinámica molecular, es a menudo posible incluir física compleja con facilidad.

Aun cuando la idea de usar partículas puede resultar natural, no siempre es posible reconocer con facilidad qué interacciones entre las partículas reproducirán fielmente las ecuaciones de la dinámica de fluidos o la mecánica del continuo. Una forma de hacer lo anterior es derivar las ecuaciones de movimiento usando una técnica de estimación de Kernel. El Kernel de suavizado o simplemente “Kernel” [2][3][8] es una función que describe cómo una distribución de densidad está asociada con una partícula de densidad  $\rho$ . Formalmente definimos a los kernels como funciones del tipo:

$$W_{ab} = W(r_a - r_b, h), \quad (3)$$

donde  $a$  es la partícula en la que está centrada la función y  $b$  es una partícula dentro del soporte compacto<sup>6</sup> de la función kernel, controlado éste último por  $h$ , la

---

<sup>6</sup>Se denomina soporte de una función al conjunto de puntos donde la función no es cero. Se

---

*smoothing length*(longitud de suavizado) que se detallará mas adelante.

Cuando aplicamos esto a la interpolación, nos arroja un estimado de una función en cualquier punto, usando los valores de la función con respecto a la posición de las partículas. Esta estimación de la función puede ser diferenciada exactamente siempre que el Kernel sea diferenciable. De este modo los gradientes requeridos para las ecuaciones de la dinámica de fluidos pueden ser escritos en término de las propiedades de las partículas.

Dado que SPH es esencialmente una técnica para aproximar ecuaciones del medio continuo, puede ser usada por una amplia gama de problemas de la dinámica de fluidos.

La interpolación mediante SPH de una cantidad  $A$ , la cual es una función de las coordenadas espaciales, está basada en la integral interpolante [8]

$$A(\mathbf{r}) = \int A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}', \quad (4)$$

donde la integral se lleva a cabo sobre todo el espacio, la función  $W$  es el Kernel, el cual tiene dos propiedades

$$\int W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = 1 \quad (5)$$

y

$$\lim_{h \rightarrow 0} W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = \delta(\mathbf{r} - \mathbf{r}'), \quad (6)$$

donde  $d\mathbf{r}'$  es el elemento diferencial de volumen. El Kernel  $W$  debe estar normalizado a uno dado que estadísticamente, la probabilidad de obtener un cierto valor específico para una cantidad física  $A$  en el intervalo finito de la integral 4 debe ser igual a uno. La interpolación reproduce la cantidad  $A$  exáctamente si el Kernel es una función delta. En la práctica, los Kernel son funciones, las cuales tienden a una función delta en la medida en que la longitud de escala  $h$  tiende a cero. Estos Kernel están normalizados a uno y, por tanto, las cantidades son interpoladas exáctamente. Un Kernel usado comúnmente en la aplicación del método SPH, y que es con el que se ha trabajado en este proyecto, es el Kernel  $M_4$ , que tiene la forma:

$$M_4(x) = \begin{cases} \frac{1}{6} [(2 - q)^3 - 4(1 - q)^3], & \text{para } 0 \leq q \leq 1, \\ \frac{1}{6} [(2 - q)^3], & \text{para } 1 \leq q \leq 2, \\ 0, & \text{para } q \geq 2, \end{cases} \quad (7)$$

---

dice que una función tiene soporte compacto si el conjunto donde no es nula conforma un conjunto cerrado y acotado.

---

con  $q = \frac{|x|}{h}$ . El Kernel SPH asociado con  $M_n(x)$  en una dimensión es  $W(x, h) = M_n(x)/h$ . En  $d$  dimensiones la misma expresión es usada pero será multiplicada por  $1/h^d$  y una constante, para asegurar que esté normalizado en el nuevo espacio. Por ejemplo, el factor  $1/6$  en el caso anterior será remplazado por  $5/(14\pi)$  en dos dimensiones y por  $1/(4\pi)$  en tres dimensiones. El Kernel  $M4$ , también llamado  $CS$  (*Cubic Spline*), ha sido por mucho, el kernel más usado y analizado en la literatura emergente que involucra al método SPH; esto, dado que se asemeja a una función gaussiana al tiempo que es de soporte compacto.

Para aplicar esta interpolación a un fluido, dividimos el mismo en un conjunto de pequeños elementos de masa. El elemento “ $a$ ” tendrá una masa  $m_a$ , una densidad  $\rho_a$  y una posición  $\mathbf{r}_a$ . El valor de  $A$  para la partícula “ $a$ ” es denotado como  $A_a$ . La integral puede entonces aproximarse mediante una sumatoria sobre los elementos de masa. Esto nos arroja la sumatoria interpolante:

$$A(\mathbf{r}) = \int A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = \sum_{j=1}^N A_j W(\mathbf{r} - \mathbf{r}_j, h) \Delta V_j,$$

$$A(\mathbf{r}) = \sum_{j=1}^N A_j W(\mathbf{r} - \mathbf{r}_j, h) \frac{1}{\rho_j} (\rho_j \Delta V_j),$$

donde el elemento de masa  $m_j$  es  $\rho_j \Delta V_j$ . Con lo que finalmente,

$$A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \quad (8)$$

donde la sumatoria se realiza sobre todas las partículas, pero en la práctica, sólo se lleva a cabo sobre los vecinos cercanos dado que  $W$  cae rápidamente con respecto a la distancia.

Lo anterior se debe a la escala  $h$ , la cual representa la distancia o radio de alcance con la cual se toman los vecinos correspondientes a una partícula  $a$ -ésima a fin de interpolar la cantidad  $A$  sobre de ella. Dada la forma del Kernel  $CS$ , el radio de alcance con el que dicha partícula tomará a sus vecinos será de  $2h^7$ .

Como un ejemplo del uso de la estimación por Kernel, supongamos que  $A$  es la densidad  $\rho$ . La interpolación nos arroja la siguiente estimación para la densidad en un punto en torno a  $\mathbf{r}$ :

$$\rho(\mathbf{r}) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h), \quad (9)$$

---

<sup>7</sup>Esto se debe a que el kernel  $CS$  tiene 3 regiones de influencia con respecto a  $q$ , la primera que va de 0 a 1 y es la que más aporta a los valores calculados, la segunda que va de 1 a 2 y que aporta un valor menor, y la tercera que va de 2 en adelante y que no aporta nada a los valores calculados

---

lo cual muestra cómo la masa del conjunto de partículas es suavizada para producir la estimación para la densidad.

## Aleatoriedad vs determinismo.

En la ecuación 8 se realizó un desarrollo algebraico en el que discretizamos la integral interpolante al considerar que dividíamos al fluido (con el que caracterizamos al sistema físico al emplear el método de SPH) en un conjunto de pequeños elementos de masa. No obstante sabemos que las posiciones  $\mathbf{r}$  y  $\mathbf{r}_j$  de las partículas (o elementos del fluido) fueron generadas mediante el método de aceptación y rechazo y, por tanto, el desarrollo hecho de las ecuaciones 4 a la 8 involucra una integral determinista calculada sobre variables aleatorias tal como en el caso de una integral de Monte Carlo.

Surge entonces la pregunta, ¿Cómo se justifica el hecho de utilizar variables aleatorias al calcular integrales deterministas? [9]. Podemos conectar estos conceptos pensando en la integral de la forma

$$I = \int_a^b f(x) dx, \quad (10)$$

Podemos ver a  $I$  como un promedio si consideramos una densidad unidimensional  $\rho(x)$ , con la cual podemos expresar a la integral como:

$$I = \int_a^b \rho(x) \left( \frac{f(x)}{\rho(x)} \right) dx \equiv \left\langle \frac{f(x)}{\rho(x)} \right\rangle, \quad (11)$$

Por tanto esta integral es en sí misma un promedio de la función  $\frac{f(x)}{\rho(x)}$ . Si ahora pensamos en aproximar este promedio mediante una discretización de puntos aleatorios  $\{x_j\}_{j=1}^{N_x}$  (esto es, posiciones de puntos, generadas mediante variables aleatorias tal como en el método de aceptación y rechazo), podremos escribir entonces:

$$I \approx \frac{1}{N_x} \sum_{j=1}^{N_x} \frac{f(x_j)}{\rho(x_j)}. \quad (12)$$

De lo cual se desprende que para una función definida sobre  $[a, b]$ , la distribución que gobierna a  $x_j \in [a, b]$  es justamente  $\rho(x) = \frac{1}{(b-a)}$ , esto dado que una variable aleatoria  $x_j$  que est distribuida uniformemente en el intervalo  $[a, b]$ , si su función de densidad es:

$$f(x_j) = \begin{cases} \frac{1}{(b-a)}, & \text{si } a \leq x_j \leq b, \\ 0, & \text{para cualquier otro valor.} \end{cases}$$

---

De acuerdo a lo anterior, una variable aleatoria distribuida uniformemente, tiene una función de densidad que es una constante en el intervalo de definición; lo cual nos lleva a la noción de una distribución aleatoria dado que nuestra función definida sobre  $[a, b]$ , describe la probabilidad relativa según la cual una variable aleatoria  $x$  tomará un determinado valor. Insertando esto en nuestra suma obtenemos finalmente:

$$I \approx \frac{1}{N_x} \sum_{j=1}^{N_x} f(x_j) (b-a) = \sum_{j=1}^{N_x} f(x_j) \Delta x, \quad (13)$$

donde  $\Delta x = \frac{(b-a)}{N_x}$ . Con lo que podemos concluir que:

$$\int_a^b f(x) dx = \sum_{j=1}^{N_x} f(x_j) \Delta x, \quad (14)$$

donde la expresión anterior se aproxima a la igualdad conforme  $N_x$  se aproxima al infinito. Entonces una distribución de probabilidad o función de densidad de probabilidad de  $x$  es una función  $f(x)$  tal que para cualesquiera dos números  $a$  y  $b$  con  $a \leq b$ , se tiene que la probabilidad de encontrar a  $x$  entre  $a$  y  $b$  está dada por:

$$P(a \leq x \leq b) = \int_a^b f(x) dx. \quad (15)$$



# Capítulo 1

## Optimización numérica multivariable sin restricciones.

### 1.1. Métodos de relajación.

Si recordamos del preámbulo de este trabajo, nuestro objetivo será optimizar los valores generados para las posiciones iniciales de un conjunto de partículas (elementos del fluido) que representan a nuestro sistema físico. Un problema de optimización consiste en maximizar o minimizar una función eligiendo sistemáticamente valores de entrada (tomados de un conjunto permitido) y calculando el valor de la función. Numéricamente existen diversos métodos para realizar un proceso de optimización. El presente trabajo se centrará particularmente en aquellos que emplean al gradiente como herramienta de optimización.

La optimización numérica de funciones no lineales requiere la utilización de técnicas de optimización eficientes y robustas. La eficiencia es importante porque la solución de estos problemas se lleva a cabo por un procedimiento iterativo. La robustez (habilidad para encontrar la solución) es una propiedad deseable, dado que el comportamiento de las funciones no lineales puede ser difícil de predecir computacionalmente: pueden presentarse máximos, mínimos y puntos de silla de forma local y global. En algunas regiones el avance hacia el punto óptimo puede ser muy lento, necesitando mucho tiempo de cálculo, etc.

La mayor parte de los algoritmos de cálculo siguen una metodología similar. Se determina un punto inicial, se evalúa la función en ese punto y se elige una dirección de búsqueda. Se comienza entonces con un movimiento en la dirección de búsqueda hasta encontrar un óptimo en esa dirección para un mismo ciclo, o bien hasta que se produzca una mejoría determinada con respecto al valor de la función.

A continuación se selecciona una nueva dirección y, así, sucesivamente.

El problema es minimizar (o maximizar) funciones de algunos vectores que a menudo tienen un gran número de dimensiones. Muchos problemas pueden formularse de esta manera: por ejemplo, un programa de ajedrez de computadora podría ser visto como un intento de encontrar el conjunto de, digamos, diez movimientos que producen la mejor jugada posible para un participante en cierto momento de una partida.

Tal como se comentó en la sección , la motivación del proyecto a realizar, radica en minimizar la dispersión obtenida del perfil de densidad conformado por los datos iniciales. Para ello se pueden emplear diversos métodos que nos permitan de algún modo llevar a cabo este proceso.

Considerando su aparente sencillez de programación y su rápida velocidad de proceso, se ha optado por trabajar con métodos basados en el descenso a lo largo de la dirección del gradiente.

## 1.2. Método del gradiente.

El método del gradiente o método de máxima pendiente [9][10][11], consiste en la búsqueda de la dirección de máximo descenso de una cierta función a optimizar (que en nuestro caso será la función de dispersión), mediante el cálculo del gradiente de dicha función; lo cual viene siendo lo mismo que encontrar el mínimo de la función.

Una buena dirección de búsqueda debería reducir (para minimización) la función objetivo, de tal manera que si  $\mathbf{x}^0$  es el punto inicial y  $\mathbf{x}^1$  es el nuevo punto, se cumpla que

$$f(\mathbf{x}^1) < f(\mathbf{x}^0). \quad (1.1)$$

Recordemos que el gradiente es un vector en un punto  $\mathbf{x}$  que proporciona la dirección (local) de máxima variación de la función. El vector gradiente es un vector ortogonal al contorno de la función del punto. Por ello, en la búsqueda de un mínimo, la dirección de movimiento será contragradiente.

$$\mathbf{S}^k = -\nabla f(\mathbf{x}^0). \quad (1.2)$$

En el método de máximo descenso, la transición de un punto  $x^k$  a otro  $x^{k+1}$  viene dada por la siguiente expresión:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k = \mathbf{x}^k + \alpha^k \mathbf{S}^k = \mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k), \quad (1.3)$$

donde:

$\Delta\mathbf{x}^k$  es el vector que va desde  $\mathbf{x}^k$  hasta  $\mathbf{x}^{k+1}$  y cuya magnitud está dada por el

gradiente de la función.

$\mathbf{S}^k$  es la dirección de búsqueda de máximo descenso.

$\alpha^k$  es un parámetro que determina la longitud de paso en la dirección de máximo descenso.

El gradiente negativo da la dirección de movimiento, pero no la longitud de dicho movimiento. Por lo tanto, existen varios procedimientos posibles dependiendo de la elección de  $\alpha^k$ . Entre los diversos métodos que existen para la selección de la longitud de un paso, dos merecen una mención especial. El primero de ellos emplea una búsqueda unidireccional de la longitud en la dirección del gradiente. El segundo especifica a priori la longitud del paso para cada iteración.

El algoritmo práctico lo podemos resumir en los siguientes pasos:

- 1.- Elegir un valor inicial  $\mathbf{x}^0$ . En pasos sucesivos será  $\mathbf{x}^k$ .
- 2.- Calcular analítica o numéricamente las derivadas parciales de  $f(\mathbf{x}^k)$ .
- 3.- Calcular el vector de búsqueda  $\mathbf{S}^k$ .
- 4.- Usar la relación de transición 1.3 de un punto a otro para obtener el siguiente punto. El valor  $\alpha^k$  puede ser de valor fijo o calculado en cada paso mediante una búsqueda unidireccional.
- 5.- Comparar  $f(\mathbf{x}^{k+1})$  con  $f(\mathbf{x}^k)$ . Si el cambio es menor que una tolerancia preespecificada terminar; en caso contrario volver a paso dos y continuar con las iteraciones.

Un método de rápido descenso puede terminar en cualquier punto estacionario; es decir, puede llegar a un mínimo local o a un punto silla. Para conocer qué tipo de resultado hemos obtenido debemos asegurarnos que la matriz hessiana está definida positiva.

### 1.3. Búsqueda lineal inexacta del paso.

Una vez calculada la dirección de desplazamiento  $\mathbf{S}$  en un cierto punto del proceso, surge la pregunta de, ¿qué paso dar a lo largo de dicha dirección?, es decir, ¿cuánto moverse en esa dirección? La solución para responder a la pregunta la encontramos en minimizar la función  $\chi(\alpha)$  en  $\alpha$  dada por:

$$\chi(\alpha) = f(\mathbf{x} + \alpha\mathbf{S}), \quad (1.4)$$

a lo cual se le llama búsqueda lineal inexacta del paso o simplemente búsqueda lineal [12]. En este método la búsqueda puede hacerse de forma inexacta con un coste en cálculos menor. En tal caso, hay que garantizar que se cumpla que

$$f(\mathbf{x} + \alpha\mathbf{S}) \neq f(\mathbf{x}), \quad (1.5)$$

y a su vez que la función decrezca suficientemente con respecto a algún criterio establecido. Para hacer esto, consideremos que la derivada de la función  $f(\mathbf{x} + \alpha\mathbf{S})$  con respecto a  $\alpha$  es

$$f'(\mathbf{x} + \alpha\mathbf{S})_{\alpha} = \frac{\partial f(\mathbf{x} + \alpha\mathbf{S})}{\partial x_1} S_1 + \frac{\partial f(\mathbf{x} + \alpha\mathbf{S})}{\partial x_2} S_2 + \dots + \frac{\partial f(\mathbf{x} + \alpha\mathbf{S})}{\partial x_n} S_n, \quad (1.6)$$

$$f'(\mathbf{x} + \alpha\mathbf{S})_{\alpha} = \nabla f(\mathbf{x} + \alpha\mathbf{S})^T \mathbf{S}. \quad (1.7)$$

Entonces, para  $\alpha = 0$ , se tendrá que

$$\begin{aligned} f(\mathbf{x} + \alpha\mathbf{S})_{\alpha=0} &= f(\mathbf{x}) \\ f'(\mathbf{x} + \alpha\mathbf{S})_{\alpha=0} &= \nabla f(\mathbf{x})^T \mathbf{S}. \end{aligned} \quad (1.8)$$

Y, así, la aproximación lineal de  $f(\mathbf{x} + \alpha\mathbf{S})$  en  $\alpha = 0$  es

$$f(\mathbf{x} + \alpha\mathbf{S})_{\alpha=0} + \alpha f'(\mathbf{x} + \alpha\mathbf{S})_{\alpha=0} = f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})^T \mathbf{S}. \quad (1.9)$$

Hecho lo anterior, establezcamos un criterio bajo el cual  $\alpha$  sea tal que la función a minimizar decrezca razonablemente. Un práctico criterio popular para determinar una búsqueda lineal es la regla de Armijo [12] (ver figura 1.1). La idea esencial es que la regla debe primero garantizar que la selección de  $\alpha$  no arroje un valor demasiado grande. El **criterio de Armijo** establece que  $\alpha$  debe cumplir con

$$f(\mathbf{x} + \alpha\mathbf{S}) \leq f(\mathbf{x}) + \rho\alpha \nabla f(\mathbf{x})^T \mathbf{S}; \quad (1.10)$$

donde  $\rho \in (0, 1)$  es una constante a elegir. Este criterio garantiza que la función decrezca razonablemente.

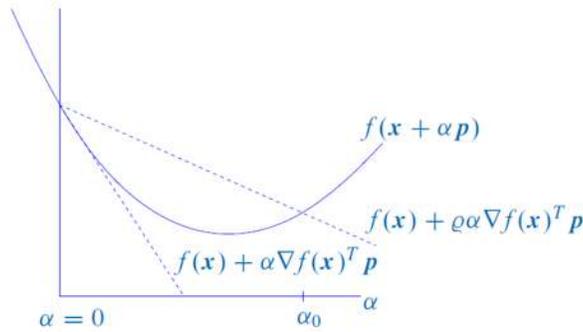


Figura 1.1: Criterio de Armijo; la sección de la curva  $f(\mathbf{x} + \alpha\mathbf{S})$ , que se encuentra por debajo de la línea punteada que representa a  $f(\mathbf{x}) + \rho\alpha \nabla f(\mathbf{x})^T \mathbf{S}$  (concretamente desde  $\alpha = 0$  hasta  $\alpha = \alpha_0$ ), corresponde al conjunto de tamaños de paso que satisfacen la desigualdad.

El método inexacto más extendido para calcular la amplitud de paso  $\alpha$  se conoce como *backtracking*. Comienza con un paso completo,  $\alpha = 1$ , y lo va reduciendo mediante un factor constante,  $\rho \cdot \alpha$ ,  $\rho \in (0, 1)$ , hasta que se cumpla el criterio de Armijo.

## 1.4. Método del gradiente conjugado.

El método del gradiente conjugado [10][11], debido a *Fletcher* y *Reeves* (1964) [13], combina las características de la convergencia cuadrática del método de las direcciones conjugadas, con las del método del gradiente. El método supone una importante mejora del método del gradiente con sólo un pequeño incremento en el esfuerzo de cálculo. El método del gradiente conjugado, esencialmente, combina la información obtenida del vector gradiente con la información acerca del vector gradiente de iteraciones previas. Lo que hace el método es calcular la nueva dirección de búsqueda utilizando una combinación lineal del gradiente en la etapa considerada y el de la etapa anterior.

Los pasos del cálculo se detallan a continuación:

- 1.- En  $\mathbf{x}^0$  (posición inicial) calcular  $f(\mathbf{x}^0)$  y calcular  $\mathbf{S}^0 = -\nabla f(\mathbf{x}^0)$ .
- 2.- Almacenar  $\nabla f(\mathbf{x}^0)$  y calcular  $\mathbf{x}^1 = \mathbf{x}^0 + \alpha^0 \mathbf{S}^0$  minimizando mediante una búsqueda unidireccional en la dirección  $\mathbf{S}^0$ .
- 3.- Calcular  $f(\mathbf{x}^1)$  y  $\nabla f(\mathbf{x}^1)$ . La nueva dirección de búsqueda es una combinación lineal de  $\mathbf{S}^0$  y  $\nabla f(\mathbf{x}^1)$ :

$$\mathbf{S}^1 = -\nabla f(\mathbf{x}^1) + \mathbf{S}^0 \frac{\nabla f(\mathbf{x}^1)^T \nabla f(\mathbf{x}^1)}{\nabla f(\mathbf{x}^0)^T \nabla f(\mathbf{x}^0)}. \quad (1.11)$$

Para la etapa k-ésima la relación es:

$$\mathbf{S}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \mathbf{S}^k \frac{\nabla f(\mathbf{x}^{k+1})^T \nabla f(\mathbf{x}^{k+1})}{\nabla f(\mathbf{x}^k)^T \nabla f(\mathbf{x}^k)}. \quad (1.12)$$

Para una función cuadrática se puede demostrar que las dos direcciones de búsqueda son conjugadas.

- 4.- Realizar el análisis de convergencia, (la función objetivo ha disminuido), y terminar el algoritmo cuando  $\|\mathbf{S}^k\|$  sea menor que alguna tolerancia preestablecida.

Algunas de las dificultades que aparecen en el método de direcciones conjugadas <sup>1</sup> también aparecen en el método del gradiente conjugado. Se puede producir una dependencia lineal de las direcciones de búsqueda durante el proceso; esto se puede eliminar reiniciando el proceso cada cuatro o cinco etapas completas.

Todos los métodos expuestos a lo largo de este capítulo, y que tienen como herramienta fundamental al gradiente de la función, serán empleados para construir un algoritmo capaz de optimizar las posiciones de las partículas tal como se describe en la sección .

---

<sup>1</sup>Ver apéndice A

# Capítulo 2

## Construcción del proceso de optimización.

### 2.1. Resultados iniciales.

Ahora que ya conocemos los métodos a implementar con el fin de optimizar las posiciones de las partículas generadas por el método de aceptación y rechazo, sólo resta adaptar estos métodos de optimización basados en el gradiente de la función a nuestro caso particular en el que las propiedades de los sistemas físicos se calculan a través de promedios vía el método SPH, lo que incluye al cálculo de gradientes de dichas propiedades físicas.

En el capítulo se habló de la importancia de los datos iniciales y del problema con el perfil de densidad generado por estos cuando tratamos de modelar un fenómeno físico. Para visualizar a detalle este problema y la necesidad intrínseca de realizar una optimización de dichos datos iniciales, consideremos un perfil de densidad que representa a cierto sistema físico (tal como un una estrella o una galaxia) dado por:

$$\rho(r) = \frac{M_T}{2\pi R^2} \frac{1}{r}, \quad (2.1)$$

donde  $M_T$  es la masa total del sistema y  $R$  el radio máximo del mismo. Dicho perfil representaría a nuestro fenómeno físico a un tiempo  $t = 0$ . Habrá que señalar que durante este trabajo se ha usado esta función como uno de nuestros caballos de batalla.

Para generar mediante el método de aceptación y rechazo a las partículas que conforman a este sistema, necesitamos obtener la función de distribución de probabilidad a partir del perfil de densidad dado. Considerando que la masa total

del sistema está dada por

$$M_T = \int \rho(r) dV, \quad (2.2)$$

donde el diferencial de volumen  $dV$  está definido como

$$dV = r^2 \sin(\theta) dr d\theta d\phi, \quad (2.3)$$

en coordenadas cartesianas, entonces

$$M_T = \int_0^{2\pi} d\phi \int_0^\pi \sin(\theta) d\theta \int_0^R \rho(r) r^2 dr, \quad (2.4)$$

de lo cual, normalizando la expresión podemos obtener las funciones de distribución deseadas para cada una de las coordenadas del sistema.

$$\begin{aligned} f(r) &= \frac{2r}{R^2}, \\ f(\theta) &= \frac{\sin(\theta)}{2}, \\ f(\phi) &= \frac{1}{2\pi}. \end{aligned} \quad (2.5)$$

Con éstas podemos obtener mediante el método de aceptación y rechazo descrito antes, el valor de cada una de las variables aleatorias  $r$ ,  $\theta$  y  $\phi$  para cada una de las  $N$  partículas que forman al sistema; donde, a su vez, dichas variables aleatorias están asociadas respectivamente con las funciones de distribución de probabilidad  $f(r)$ ,  $f(\theta)$ , y  $f(\phi)$ . El resultado de aplicar el proceso antes descrito se muestra en la figura 2.1 (en coordenadas cartesianas).

Para corroborar que las partículas obtenidas conforman un perfil de densidad aproximado al perfil de densidad deseado, se recurrirá a la ecuación 9. Esto nos proporcionará un promedio de la densidad con respecto a la posición de cada partícula, a través del cual se derivará el perfil de densidad de las partículas generadas mediante el método de aceptación y rechazo. Donde el kernel M4 antes descrito, en el caso de 3D, toma la forma:

$$W(q, h) = \begin{cases} \frac{1}{4\pi h^3} [(2-q)^3 - 4(1-q)^3], & \text{para } 0 \leq q \leq 1, \\ \frac{1}{4\pi h^3} [(2-q)^3], & \text{para } 1 \leq q \leq 2, \\ 0, & \text{para } q \geq 2, \end{cases} \quad (2.6)$$

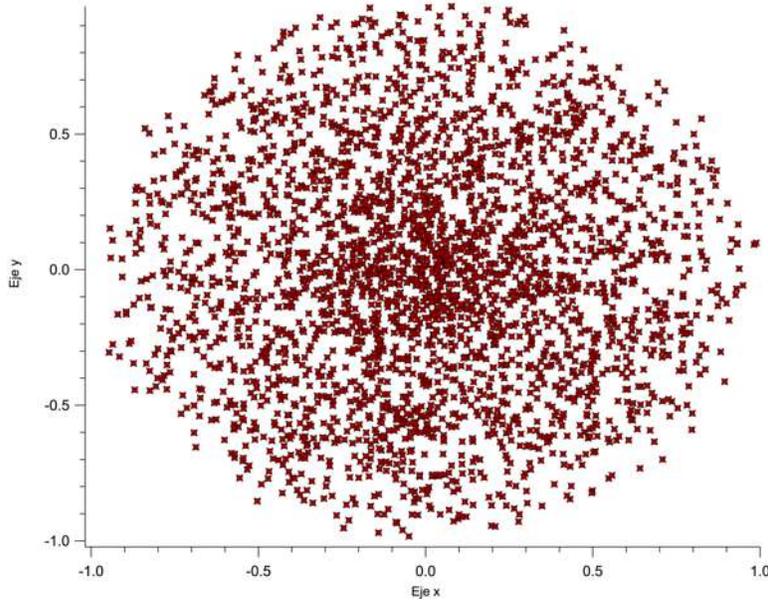


Figura 2.1: Posiciones de las partículas generadas mediante el método de aceptación y rechazo (en coordenadas cartesianas) para el sistema descrito en la ecuación 2.1.

con  $q = \frac{|x|}{h}$ , donde  $|x|$  es la norma o distancia euclidiana entre la partícula  $i$ -ésima y la  $j$ -ésima.

Cada uno de los promedios para la densidad será calculado sobre la partícula  $i$ -ésima con respecto a todos sus vecinos “influyentes”, donde la responsabilidad de determinar qué partículas vecinas nos ayudan a calcular un buen promedio para dicha densidad  $i$ -ésima, recae puramente en el Kernel y en la longitud de escala  $h$ . La forma del Kernel  $CS$  nos permitirá delimitar las regiones de influencia de las partículas, las que estén más cerca tendrán más peso en la densidad promedio mientras que las más lejanas no aportarán nada; así, entonces, para el cálculo promedio de la densidad con respecto a la partícula  $i$ -ésima, será justo ella misma quien más aporte al cálculo de dicho promedio.

Por su parte,  $h$  nos permitirá ajustar la distancia de influencia a fin de calibrar nuestras aproximaciones. La longitud de escala  $h$  está dada por la ecuación

$$h = \eta \left( \frac{m}{\rho(\mathbf{r})} \right)^{\frac{1}{3}}, \quad (2.7)$$

donde  $\eta$  es una constante que se ajusta generalmente a 1.5, aunque habría que señalar que para este trabajo funcionó mejor con valores ligeramente inferiores, a la par es importante recordar que para el Kernel  $CS$  (empleado en este trabajo), la distancia de influencia para calcular los promedios a la SPH es de  $2h$ . Nótese que en

la fórmula para la longitud  $h$  se usan como parámetros a la masa  $m$  de la partícula sobre la que se calcula el promedio y a la densidad analítica  $\rho(r)$  en ese punto; estos valores son datos iniciales con los que ya contamos, o que podemos fijar sin problemas. En este trabajo, la masa de las partículas es la misma para todas ellas (que es lo que se hace comúnmente por convención cuando se trabaja con SPH) y en valor es igual a  $M_T/N$ , con  $N$  igual al número de partículas. Por su parte la densidad analítica con respecto a la partícula, depende intrínsecamente de la posición en la que esté colocada la partícula en cuestión.

De lo anterior, dado que ya contamos con la expresión para la densidad analítica  $\rho(r)$  del sistema, puesto que hemos fijado el valor para las masas  $m$  de las partículas y ya que hemos generado las posiciones de las partículas, podemos sin problema calcular  $h$  y  $q$  con lo cual, a su vez, podemos hacer el cálculo de los promedios de la densidad para obtener el perfil numérico correspondiente, para así poderlo comparar con el perfil analítico respectivo. El resultado de esta comparación se muestra en la imagen 2.2:

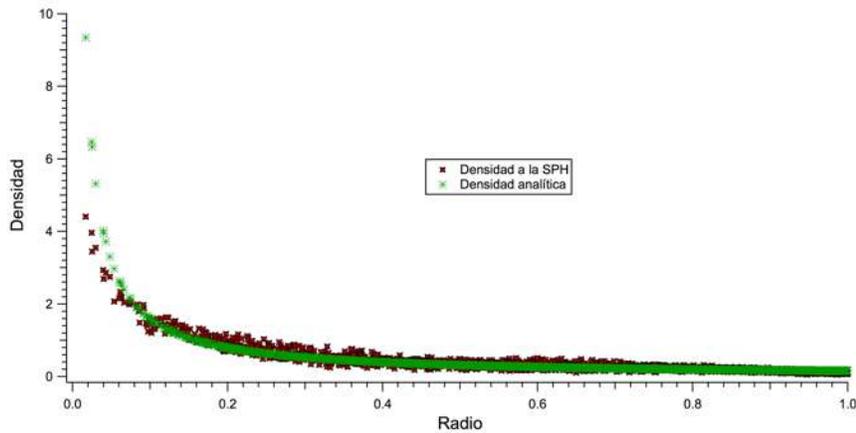


Figura 2.2: Comparativa del perfil de densidad a la SPH para las posiciones iniciales de los puntos generados mediante el método de aceptación y rechazo con respecto al perfil de densidad analítico en el caso del sistema asociado con la ecuación 2.1.

En la figura 2.2, la curva más delgada, formada por asteriscos de color verde, representa al perfil analítico generado numéricamente y la curva más ancha es la que representa al perfil numérico a la SPH. De la gráfica se puede notar que, aunque nuestros datos iniciales se aproximan al valor correcto, tienen un amplio error con respecto al mismo. Dado que el método de aceptación y rechazo es, tal como se dijo antes, un método tipo Monte Carlo, el error antes mencionado será del orden de  $\frac{1}{\sqrt{N}}$ .

La idea de emplear un método de relajación recae justamente en reducir este error a fin de que nuestros datos iniciales sean lo más viables posibles para su posterior implementación en la evolución y estudio del fenómeno físico deseado en trabajos futuros.

## 2.2. Construcción del método.

A fin de armar el código que usaremos para la optimización de los datos iniciales mediante método del gradiente, debemos considerar que dado que trabajamos con  $n_p$  partículas, nuestro sistema tendrá  $3N$  grados de libertad. Luego entonces, el método del gradiente modificará cada una de las  $3N$  variables del sistema en una dirección contragradiante a fin de encontrar el mínimo de la función deseada.

Pero, ¿cuál es esa función a minimizar? Para responder a esa pregunta, recordemos que nuestra meta, al aplicar el método del gradiente, es poder reducir el error en las posiciones de las partículas generadas mediante el método de aceptación y rechazo para un perfil de densidad dado, en comparación con el valor analítico de dicho perfil.

Dicho de otro modo, deseamos que el método del gradiente mueva a cada una de las partículas a una posición ‘mejorada’, mediante la cual el perfil de densidad numérico sea aún más cercano al perfil de densidad analítico  $\rho(r)$ . Así, lo que debemos minimizar es una cierta función error que propondremos como:

$$E = \frac{1}{2} \sum_{i=1}^{n_p} (\rho(\mathbf{x}_i) - \langle \rho_i \rangle)^2, \quad (2.8)$$

donde  $n_p$  es el número total de partículas,  $\rho(\mathbf{x}_i)$  es la función de densidad analítica evaluada en la partícula  $i$ -ésima y  $\langle \rho_i \rangle$  es la densidad promediada a la SPH para la partícula  $i$ -ésima. El factor  $\frac{1}{2}$  se agrega por conveniencia operacional.

El mínimo de esta función error lo encontraremos cuando se cumpla la condición,

$$(\rho(\mathbf{x}_i) - \langle \rho_i \rangle) = 0, \quad (2.9)$$

lo cual ocurrirá cuando nuestro valor para la densidad promediada a la SPH sea igual al valor de la densidad analítica.

Habrá que denotar que aun cuando la idea para optimizar nuestros datos iniciales es mover las partículas de su posición inicial a una posición óptima’, matemáticamente hablando el método del gradiente simplemente encontrará el mínimo de una función que existe en un espacio  $3N$  dimensional. Esto se ve reflejado en el hecho

de que realmente estaremos modificando iteración a iteración el valor de cada una de las  $3N$  componentes que conforman a un vector en ese espacio  $3N$  dimensional con el fin de llevar a ese vector hasta el mínimo de la función.

Notemos a su vez que esta función a minimizar es una función compuesta, en la que cada una de las variables que la componen están acopladas entre sí, debido a que el promedio de la densidad calculado a la SPH requiere no sólo del valor de la posición de la partícula sobre la que se calcula dicho promedio, sino que también del valor de la posición de sus vecinos. Esto es, a medida que modifiquemos la posición de alguna de las partículas a fin de que su densidad promedio a la SPH se ajuste mejor a la de la densidad analítica, estaremos indirectamente modificando también el valor de la densidad promedio a la SPH de sus partículas vecinas.

Debido a lo anterior, será importante asegurarse de que las modificaciones hechas para cada una de las componentes de la función a minimizar, sean aplicadas al mismo tiempo y no una por una. Esto es, el cambio en la posición de cada una de las partículas deberá ser calculado y aplicado al mismo tiempo para todas; calcular el cambio en la posición para una sola de ellas y luego modificar su posición, implicaría que al calcular el cambio en la posición para una subsecuente partícula vecina, se estaría considerando un diferente sistema al inicial.

Para determinar el cambio a aplicar en las posiciones de cada una de las partículas, será necesario conocer el gradiente de la función error sobre la partícula  $i$ -ésima; esto es,

$$\nabla_k E = \nabla_k \left[ \frac{1}{2} \sum_{i=1}^{n_p} (\rho(\mathbf{x}_i) - \langle \rho_i \rangle)^2 \right], \quad (2.10)$$

donde el gradiente actúa sobre las partículas  $k$ -ésimas. Despejando y derivando obtenemos la expresión

$$\nabla_k E = \frac{2}{2} \sum_{i=1}^{n_p} [\rho(\mathbf{x}_i) \nabla_k [\rho(\mathbf{x}_i)] - \rho(\mathbf{x}_i) \langle \nabla_k \rho_i \rangle - \langle \rho_i \rangle \nabla_k [\rho(\mathbf{x}_i)] + \langle \rho_i \rangle \langle \nabla_k \rho_i \rangle]. \quad (2.11)$$

Calculemos entonces, los gradientes necesarios con respecto a la partícula  $k$ -ésima.

a) El gradiente de la densidad analítica.

$$\nabla_k [\rho(\mathbf{x}_i)] = \nabla_k \left[ \frac{M_T}{2\pi R^2} \frac{1}{(x_i^2 + y_i^2 + z_i^2)^{\frac{1}{2}}} \right]. \quad (2.12)$$

Lo cual claramente sólo es diferente de 0 cuando  $i = k$ , por tanto

$$\nabla_k [\rho(\mathbf{x}_i)] = \left[ \frac{M_T}{2\pi R^2} \frac{\mathbf{x}_i \delta_{ik}}{(x_i^2 + y_i^2 + z_i^2)^{\frac{3}{2}}} \right]. \quad (2.13)$$

b) El gradiente de la densidad promedio a la SPH [7] es

$$\langle \nabla_k \rho_i \rangle = \sum_{j=1}^{n_p} m \nabla_k W(\mathbf{x}_i - \mathbf{x}_j, h_i), \quad (2.14)$$

donde el índice  $j$  corre sobre todos los vecinos de la partícula  $i$ -ésima. Consideremos que,

$$q_{ij} = \frac{|\mathbf{x}_i - \mathbf{x}_j|}{h_i},$$

y definiendo

$$W_{ij} = W(q_{ij}),$$

tendremos,

$$\nabla_k W_{ij} = \frac{\partial W(q_{ij})}{\partial q_{ij}} \frac{\partial q_{ij}}{\partial \mathbf{x}_k}, \quad (2.15)$$

donde

$$\frac{\partial q_{ij}}{\partial \mathbf{x}_k} = \frac{1}{h_i} \frac{(\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|} (\delta_{ik} - \delta_{jk}),$$

y entonces

$$\langle \nabla_k \rho_i \rangle = \sum_{j=1}^{n_p} \frac{m}{h_i} \frac{\partial W(q_{ij})}{\partial q_{ij}} \frac{(\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|} (\delta_{ik} - \delta_{jk}). \quad (2.16)$$

Considerando ahora que para una función analítica como en el caso a) sólo para  $i = k$  se debe cumplir que

$$\frac{\partial \Theta_i}{\partial x_k} \neq 0, \quad (2.17)$$

entonces, nos es claro que el índice  $k$  debe ser igual al índice  $i$  ó al índice  $j$ , y el hecho de tener ambas deltas en la ecuación 2.16, sólo nos está marcando la orientación de la derivada. Por tanto, considerando  $i = k$ , tendremos que

$$\langle \nabla_k \rho_k \rangle = \sum_{j=1}^{n_p} \frac{m}{h_k} \frac{\partial W(q_{kj})}{\partial q_{kj}} \frac{(\mathbf{x}_k - \mathbf{x}_j)}{|\mathbf{x}_k - \mathbf{x}_j|}. \quad (2.18)$$

Así, regresando a nuestro desarrollo del gradiente de la función error con respecto a la partícula  $k$ -ésima, sustituyendo los recientes resultados y contrayendo los índices obtendremos finalmente que

$$\nabla_k E = \left[ \rho(\mathbf{x}_i) - \sum_{l=1}^{n_p} m W_{kl} \right] \left[ \frac{M_T}{2\pi R^2} \frac{\mathbf{x}_k}{(x_i^2 + y_i^2 + z_i^2)^{\frac{3}{2}}} - \sum_{j=1}^{n_p} \frac{m}{h_k} \frac{\partial W(q_{kj})}{\partial q_{kj}} \frac{(\mathbf{x}_k - \mathbf{x}_j)}{|\mathbf{x}_k - \mathbf{x}_j|} \right], \quad (2.19)$$

donde se imponen diferentes índices  $l$  y  $j$  para reflejar el hecho de que ambas sumatorias no corren al mismo tiempo.

Con este resultado ya estamos en capacidad de llevar a cabo el algoritmo descrito en la sección 1.2 para la implementación del método del gradiente.

Lo último a ajustar en nuestro código será la condición para parar el algoritmo. Para esto hay diferentes opciones, entre las cuales está la antes mencionada de comparar  $f(\mathbf{x}_{k+1})$  con  $f(\mathbf{x}_k)$  para ver si la diferencia es menor que cierta tolerancia y entonces parar; esta opción no se aplicó dado que el método converge lentamente hacia una precisión deseada, dado que las partículas se mueven cada vez menos a medida que el perfil de densidad calculado a la SPH se acerca a su contraparte analítica.

Es por ello más útil monitorear el momento en que las partículas no cambian su posición dentro de cierta tolerancia; o dicho de otro modo, el momento en que las componentes de la función error no se modifican más con respecto a un valor de tolerancia.

Para el análisis de los diferentes casos con que se trabajó en este proyecto, muchas veces, simplemente se dejó que el código trabajase durante un cierto número de iteraciones que se consideró conveniente; esto, considerando un valor para el desplazamiento promedio de las partículas bajo el cual se notara una clara tendencia de reducción en el error absoluto. Para la descripción futura de los resultados, se indicará el tipo de condición de parada que se usó para la implementación del método en cada uno de los diversos casos analizados.

## 2.3. Análisis preliminar del método.

En la presente sección se presentarán diversos análisis realizados con el fin de corroborar el correcto funcionamiento del código desarrollado para el método del gradiente adaptado a nuestro problema de relajación de los datos iniciales. Dichos análisis, así mismo, fueron realizados para calibrar de forma idónea los parámetros que emplea el código.

Como primera aproximación, a fin de realizar estos análisis de forma sencilla, se recurrió a un perfil de densidad constante  $\rho(\mathbf{x}) = 1$  en dos dimensiones, con un área perimetral de  $3 \times 3$  unidades de longitud. Así, tendremos una configuración con forma de tablero de ajedrez en la que las partículas estarán ubicadas en cada uno de los vértices de la cuadrícula. Obsérvese que para esta configuración completamente uniforme, no se recurre en ningún momento al método de aceptación y rechazo.

Con ello, habiendo fijado la densidad y el volumen del sistema, la masa total del mismo será igual a 9 unidades. La masa de cada una de las partículas (que tal como se indicó antes, será igual para todas), estará dada por  $m = \frac{M_T}{N}$ . El número de partículas por su parte, será un parámetro a definir para cada configuración que se desee estudiar, considerando de ante mano el arreglo en forma de cuadrícula que que se impone a la configuración.

Así, si, por ejemplo, usásemos una configuración de 1296 partículas (36 partículas por lado), encontraríamos la configuración inicial mostrada en la figura 2.3

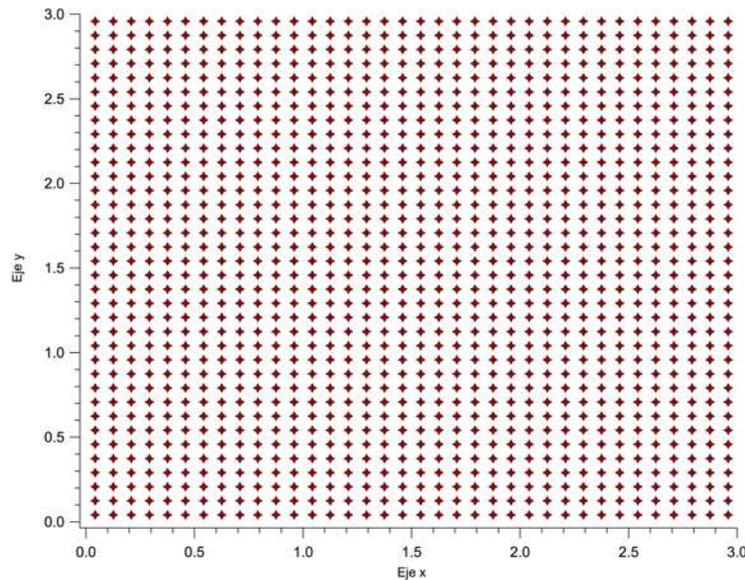


Figura 2.3: Configuración de datos iniciales caracterizada por 1296 partículas ordenadas en una cuadrícula de 36x36 posiciones.

Para dicha configuración, bien ordenada, el perfil de densidad debe ser prácticamente uno, lo cual se puede ver en la figura 2.4.

Nótese que para el perfil de densidad el valor promedio decae en los extremos, esto se debe al menor número de partículas vecinas, lo cual genera un valor promedio diferente sobre las partículas ubicadas cerca de las fronteras de la cuadrícula. Del mismo modo en el centro de la cuadrícula, el perfil es prácticamente uno, lo cual es lo que se busca. El hecho de que el promedio anterior reproduzca el valor real se

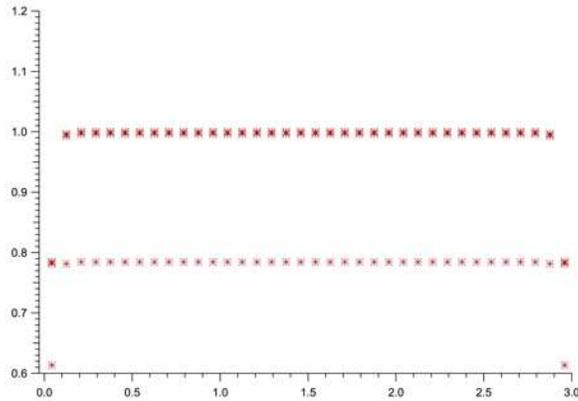


Figura 2.4: Perfil de densidad calculado a la SPH de una configuración ordenada de datos iniciales con densidad analítica constante igual a 1. Los puntos en las fronteras (que en la gráfica son los puntos por debajo del valor deseado) tienen un valor que decae debido a la falta de vecinos en la frontera que les permita calcular un buen promedio.

debe a que las partículas lejos de las fronteras tienen un buen número de vecinos que le permiten calcular un promedio correcto.

Del mismo modo el perfil de la derivada de la densidad a la SPH tendrá un comportamiento que se detalla en la figura 2.5.

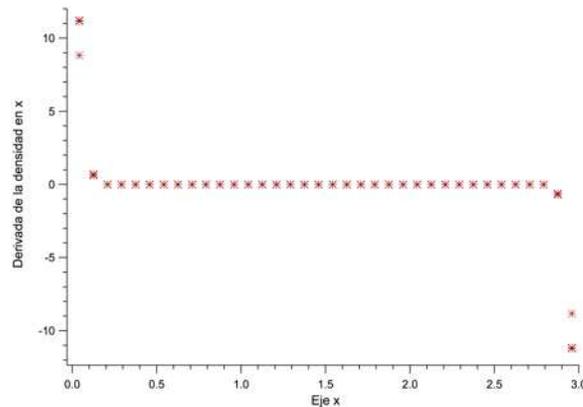


Figura 2.5: Perfil del promedio de la derivada de la densidad calculado a la SPH de una configuración ordenada de datos iniciales con densidad analítica constante igual a 1. El comportamiento en las fronteras se debe a la caída del perfil de densidad en la misma zona.

Para el caso del perfil de la derivada, se puede resaltar que dicho perfil es prácticamente 0 para las partículas lejos de las fronteras, justamente porque el perfil de densidad promediado a la SPH para éstas es prácticamente idéntico al perfil analítico. Cerca de las fronteras el perfil de la derivada decrece del infinito hasta 0 conforme nos movemos sobre el eje  $x$  desde el origen, debido a que el perfil de densidad es más cercano a uno conforme nos movemos en dicha dirección; del mismo modo, dicho perfil de la derivada crece de menos infinito a 0, conforme nos movemos sobre el eje  $x$  desde la frontera más alejada del origen hacia el centro.

Lo anterior nos deja claro que para el proceso de relajación tendremos un problema con los promedios calculados en las fronteras; y, por tanto, un problema a la hora de calcular el valor del paso a dar por cada una de las partículas cercanas a la frontera durante el proceso de relajación mediante el método del gradiente.

Para resolver este problema consideraremos una sección fronteriza de la cuadrícula para la cual las partículas se mantendrán fijas en su posición inicial durante el proceso de relajación. Así, si consideramos una sección fronteriza de 0.5 de ancho, tendremos una sección a relajar de  $2x2$ ; entonces de nuestras 1296 partículas totales, 576 caerán en la sección a relajar y el resto estarán en la sección fronteriza. Si ahora generamos las posiciones de esas 576 partículas centrales mediante el método de aceptación y rechazo, el resultado será el mostrado en la figura 2.6.

La primera prueba que nuestro código debe pasar es que para una configuración bien ordenada, es decir, el caso en que el perfil de densidad calculado a la SPH es prácticamente idéntico al perfil analítico; las partículas no deberán moverse prácticamente nada con respecto a su posición inicial. Realizando el proceso con la sección fronteriza de partículas en posiciones fijas, encontramos justamente este comportamiento donde la diferencia entre el valor inicial de las posiciones de las partículas y el valor relajado por el método del gradiente se mantiene oscilando en torno a  $1 \times 10^{-16}$ , y por tanto, el código no modifica la posición de partículas cuya configuración inicial está ya ordenada.

Para asegurar un buen comportamiento del proceso de relajación será importante que el código no tenga una dirección preferente de movimiento; esto es, que las partículas no tiendan a moverse más rápido en una dirección que en otra. Para analizar esto consideremos la configuración de 1296 partículas con fronteras fijas y 576 partículas a relajar. Estas 576 partículas a relajar las colocaremos ordenadas de forma compacta en una región confinada al centro de la cuadrícula.

Al echar a andar el proceso de relajamiento, tal como se observa en el cuadro 2.1, se produce un reacomodo de las partículas centrales, a modo que estas van

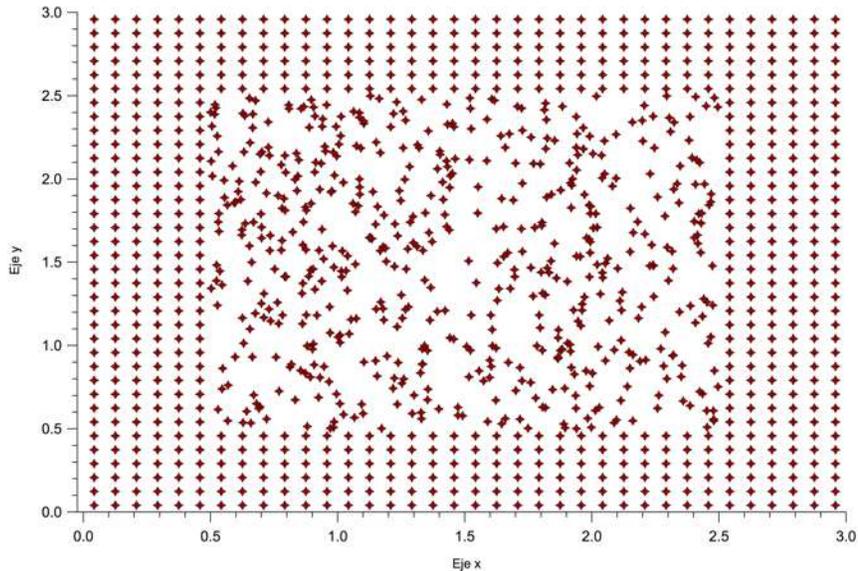
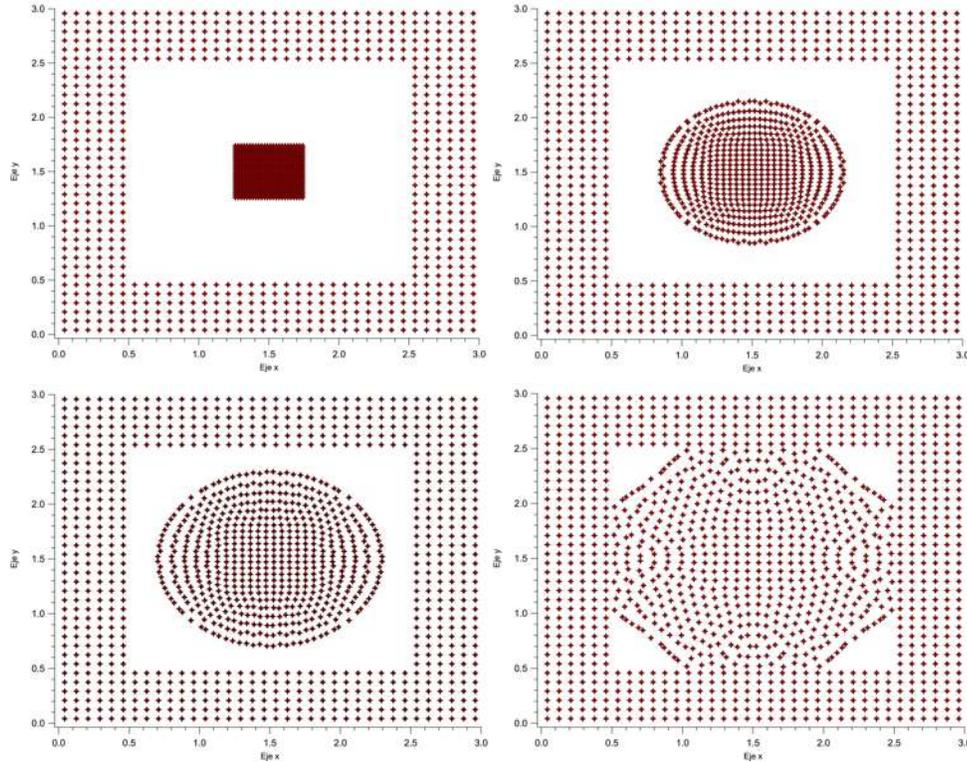


Figura 2.6: Configuración inicial de 576 partículas aleatorias a relajar generadas mediante el método de aceptación y rechazo con 720 partículas como frontera fijas. El proceso de relajación mediante el método del gradiente se ejecutará sólo sobre las partículas aleatorias.

siendo empujadas hacia las fronteras pero sin perder en cierta medida su orden inicial, dado que no existe ninguna dirección preferente en el cambio en la posición de las partículas a cada iteración.

Tal como se observa en el cuadro 2.1, conforme el proceso avanza las partículas van modificando su posición a tal grado que cada vez se acercan más al perfil de densidad deseado. Por tanto, es lógico pensar que de ser así, la función error debería estar disminuyendo progresivamente con una tendencia clara a 0, que es justamente lo que se ve en la figura 2.7 cuando graficamos el valor de dicha función a cada iteración, y que a su vez, es el valor deseado para la función error tras finalizar el proceso de relajamiento.

Una tercera prueba aplicada sobre el código consiste en comprobar que él mismo sea capaz de relajar a un conjunto prácticamente ordenado de partículas. Para esto recurriremos a un modelo sencillo de 144 partículas, en el cual solo 4 de éstas son aleatorias y el resto están ya ordenadas y fijas en su posición. El resultado de aplicar el proceso de relajación (presentado en el cuadro 2.2) nos arroja que tras un total de 5 millones iteraciones, el sistema se encontrará prácticamente relajado, oscilando en



Cuadro 2.1: Proceso de relajación para 576 partículas ordenadas y compactadas en el centro de la cuadrícula. Con una frontera de 720 partículas ordenadas fijas. De izquierda a derecha y de arriba hacia abajo se observa como las partículas aleatorias van modificando su posición al someterse al proceso de optimización sin que se observe una dirección preferente de desplazamiento.

torno a su posición final con una amplitud de desplazamiento de aproximadamente  $1 \times 10^{-7}$ .

Para el punto en que el código se detiene, el valor de la función error fue de  $2.132 \times 10^{-6}$  (ver figura 2.8). El proceso aún se podría mantener hasta llegar a una mayor precisión (un valor más cercano a 0), no obstante, el tiempo de cómputo necesario para reducir en un orden este valor, implicaría realizar cerca de 5 millones de iteraciones más. Lo anterior es uno de los primeros indicios acerca del tiempo <sup>1</sup> que requiere el código para relajar a las partículas hasta una precisión deseada.

<sup>1</sup>Dicho tiempo varió de una configuración a otra. En los casos en 2D, solo se requirió una máximo de un día de procesamiento para una configuración de 4000 partículas empleando el método del gradiente; en cambio para los casos en 3D, tanto para la densidad constante como para la densidad radial, este tiempo osciló en torno a los 4 días.

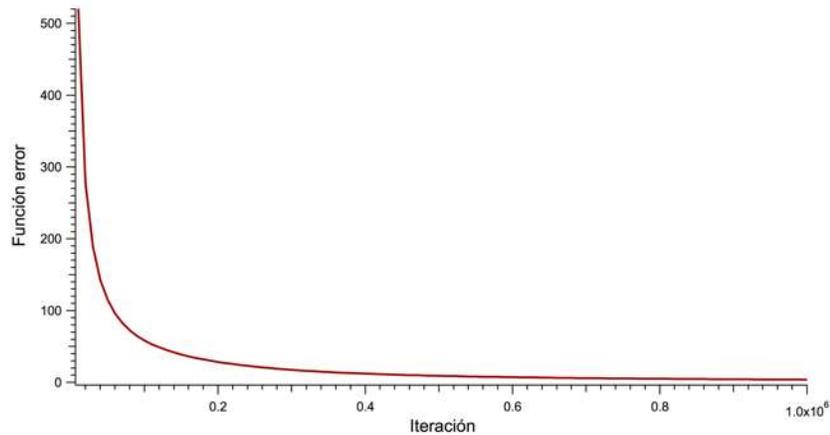


Figura 2.7: Tendencia de la función error durante el proceso de relajación para una configuración de 576 partículas a relajar.

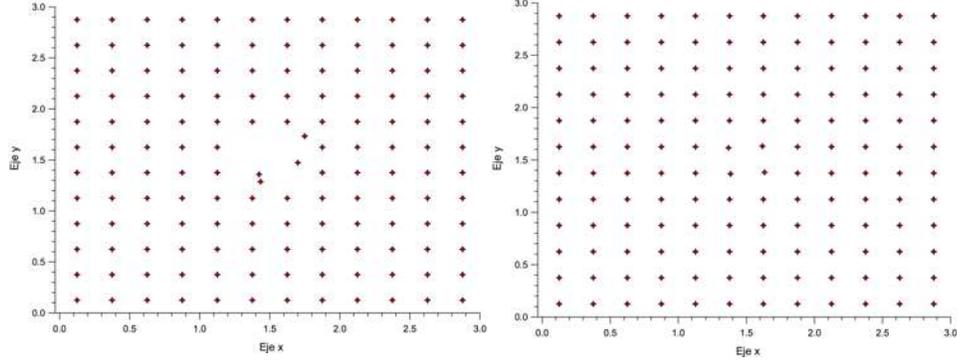
## 2.4. Parámetros iniciales.

Una vez corroborado el correcto funcionamiento del código para estos simples modelos, antes de continuar, será importante mencionar que para el monitoreo del código, durante su proceso se analizaron diferentes valores, los cuales se describen a continuación:

- El error absoluto entre el perfil de densidad analítico y el calculado mediante promedios a la SPH.
- El valor de la función error descrita en secciones previas.
- El valor promedio absoluto de los desplazamientos de las partículas.

Hay que mencionar que recurrentemente se usó el valor promedio absoluto de los desplazamientos como valor a monitorear, a fin de detener el código. Esto dado que el código, en muchas ocasiones, llegaba a entrar en un periodo en el que las partículas se desplazaban poco antes de que se encontrasen con una buena precisión para el valor de la función error. Así, una buena condición de parada será aquella en la que se pueda decir que las partículas ya no presentan desplazamientos en comparación con una cierta tolerancia.

A su vez, será importante mencionar que durante el presente análisis se fueron ajustando los parámetros que emplea el código, a fin de alcanzar su mejor desempeño. Así, se ajustó el valor de  $\eta$  a 1.2 pues se encontró un mejor comportamiento para dicho caso en comparación con los resultados que se encontraron al aplicar valores de  $\eta$  distintos. Esto se debe a que, si el valor de  $\eta$  es mas pequeño, nuestra longitud de suavizado no tomará suficientes partículas como para calcular un promedio correcto; y si es mas grande, estaremos sobre-suavizando el promedio y por tanto éste



Cuadro 2.2: Proceso de relajación para 4 partículas aleatorias ubicadas en el centro de la cuadrícula. La figura a la izquierda muestra la posición de las mismas antes de aplicar el proceso de relajación y la figura a la derecha muestra a la optimización de las posiciones. Las partículas al tener muchos vecinos ya ordenados y fijos, obtienen un buen promedio para su densidad y para la derivada de la misma.

tendrá también un valor incorrecto. Todo esto es importante pues en general en la literatura actual, para calcular promedios a la SPH se usa un valor para  $\eta$  igual a 1.5.

La longitud  $h$  de interacción con la que se contabiliza al número de vecinos a usar para cada promedio se dejó en general como un valor fijo durante el proceso de relajación, debido a que se encontró a lo largo del desarrollo de este trabajo que mantenerla fija siempre daba un mejor resultado, en comparación con tener una  $h$  variable. Para reflejarlo, se hará mención más adelante a algunos de los diferentes resultados obtenidos usando tanto una  $h$  variable como una  $h$  fija.

La  $h$  fija implica que todas las partículas tendrán la misma  $h$  durante todas las iteraciones del proceso. Dicha  $h$  emplea el perfil de densidad constante al que se desea llegar como parámetro a usar durante su cálculo. Esto es,  $h$  estará dada por:

$$h = \eta \left( \frac{m}{1.0} \right)^{\frac{1}{3}}. \quad (2.20)$$

Para el caso de la  $h$  variable, se toma la ecuación anterior sólo para la primera iteración, a fin de tener un valor inicial para la  $h$  de cada partícula. Una vez que se hace la modificación en la posición de cada una de las partículas y se calcula el valor de su promedio actual a la SPH, se usará este resultado como parámetro para calcular el valor de la  $h$  a usar por cada una de las partículas durante la siguiente iteración. Esto es, se hará uso de la ecuación:

$$h_i = \eta \left( \frac{m}{\rho(\mathbf{r})_i} \right)^{\frac{1}{3}}. \quad (2.21)$$

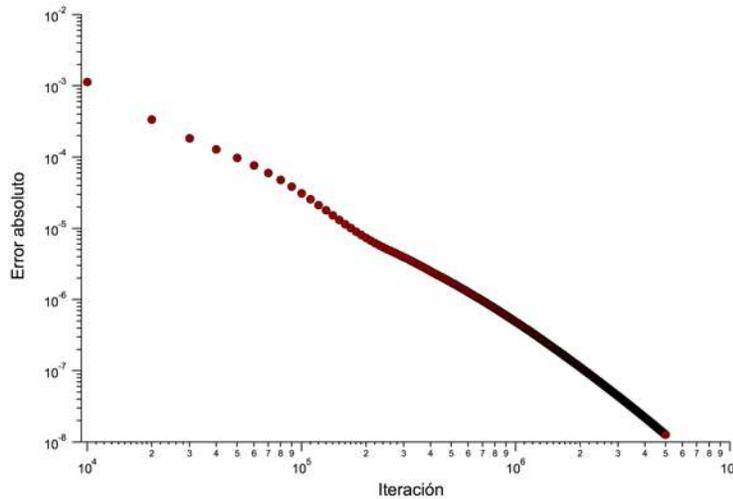


Figura 2.8: Tendencia de la función error durante el proceso de relajación en escala logarítmica para un configuración de 4 partículas a relajar. La curva muestra el rápido descenso en la magnitud del desplazamiento promedio de las partículas. La rápida caída en la curva se debe puramente al método del gradiente aplicado al método SPH; este patrón se verá repetidamente mostrado en los resultados presentados a lo largo de este trabajo.

El valor de la tolerancia para detener el código bajo el criterio de un mínimo desplazamiento de las partículas, cuando llegó a usarse, fue de  $1 \times 10^{-10}$  o menor; esto considerando que dada la tendencia del error a reducirse (ver figura 2.8), una vez que el promedio del desplazamiento de las partículas sea numéricamente pequeño, se necesitará de un largo tiempo de procesamiento extra, para obtener una mejor optimización de las posiciones iniciales de las partículas con respecto a la optimización que ya se ha conseguido..

El parámetro  $\alpha$ , que es el parámetro con el que se modifica el empuje dado por el gradiente de la función error a las partículas, se mantuvo siempre menor a 0.01, y en general fue distinto según la configuración inicial y la versión del método del gradiente que se estuviese empleando. Lo anterior recae en que una alta  $\alpha$  puede ocasionar que las partículas simplemente se muevan demasiado, provocando malos valores para los promedios, y en que una  $\alpha$  corta puede ocasionar que las posiciones de las partículas se modifiquen demasiado poco, provocando que el proceso de relajación sea por demás lento.

Una última condición sobre los parámetros iniciales consiste en tener simetrizadas nuestras ecuaciones que calculan los valores medios a fin de que, si una partícula  $i$  ve a una partícula  $j$  a la hora de calcular cierto promedio, la partícula  $j$  también la vea al calcular el suyo. Para hacer esto se puede recurrir a cualquiera de las siguientes

simetrizaciones:

$$\begin{aligned} h_{ij} &= \frac{h_i + h_j}{2} \\ \nabla_i W_{ij} &= \frac{\nabla_i W(r_{ij}, h_i) + \nabla_i W(r_{ij}, h_j)}{2}. \end{aligned} \quad (2.22)$$

En la práctica, dado que en general la longitud  $h$  se mantuvo fija, dicha simetrización en la mayoría de los casos no fue necesaria pues la magnitud de la longitud de suavizado usando  $h$  fija es idéntica para todas las partículas al tener una densidad analítica constante. Más aún, cuando se probó con el fin de mejorar los resultados en que se usaba una  $h$  variable, no se encontraron resultados contundentes que nos llevaran a considerar que usarla era la mejor opción para este caso, puesto que dichos resultados obtenidos fueron aproximadamente los mismos (las posiciones finales de cada configuración tras el proceso de relajación no mostraban diferencias claras entre un caso y el otro).

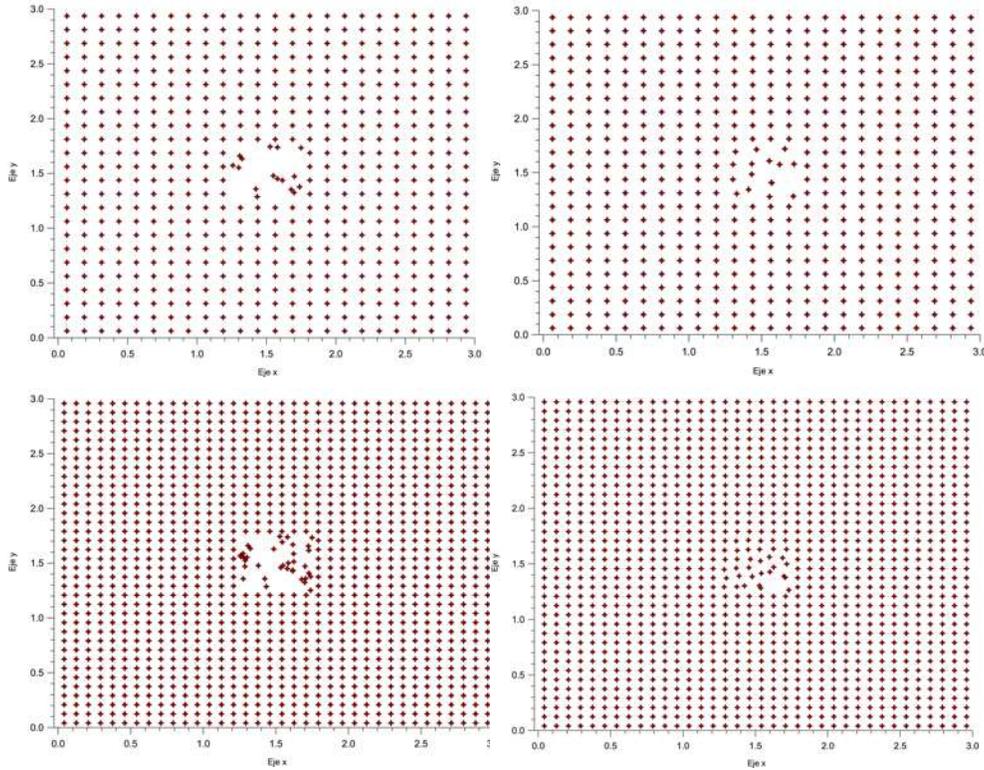
## 2.5. El problema de los pares.

Durante el análisis preliminar del método, se encontró una complicación a la hora de intentar encontrar la mejor precisión posible para el mismo en un tiempo razonable; esto es, el proceso seguía andando siempre que, bajo la misma tolerancia antes mencionada para el promedio en el desplazamiento, las partículas se mantuvieran en movimiento.

Para reflejar esta complicación, analicemos primero configuraciones en las que las partículas aleatorias son pocas y están centradas en la cuadrícula, mientras que, por tanto, las partículas vecinas a éstas serán considerablemente más numerosas y estarán fijas.

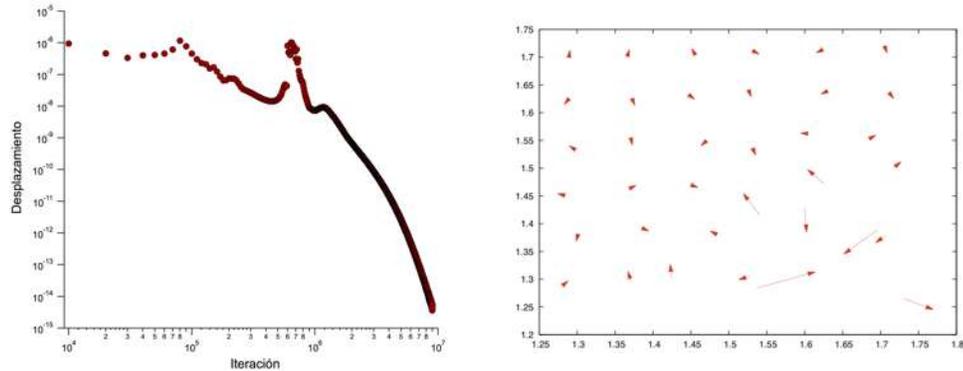
La primera de las configuraciones a analizar está conformada por 16 partículas aleatorias cuya posición se desea optimizar y que está conformada por un total de 576 partículas, y la segunda por 36 partículas aleatorias cuya posición se desea optimizar y que está conformada por un total de 1296 partículas; dichas configuraciones se presentan en el cuadro 2.3.

Tal como se puede ver, si consideramos lo mencionado antes respecto a una tolerancia en el desplazamiento de las partículas de  $1 \times 10^{-11}$  o menor, entonces nos encontraremos que por algún motivo la configuración ya no se está modificando lo suficiente con respecto a una relación entre el tiempo de procesamiento y la mejora en el resultado.



Cuadro 2.3: Proceso de relajación para partículas aleatorias centrales rodeadas por una amplia frontera de partículas fijas. A la izquierda se muestran las configuraciones iniciales de 16 (panel superior) y 36 (panel inferior) partículas aleatorias; a la derecha se muestra el resultado de aplicar el proceso de relajación. Tras el proceso de relajación observan partículas que se juntan formando pares afectando los promedios calculados a la SPH.

La primera cuestión a analizar ya que sabemos que las partículas sí se mueven pero que lo hacen en una cantidad muy pequeña, es saber en qué dirección se mueven dichas partículas puesto que eso nos permitirá saber si las partículas se están desplazando aún hacia un mejor acomodo, o si están oscilando en torno a su actual posición. Lo encontrado se muestra en el cuadro 2.4 que ejemplifica el caso en que tenemos 36 partículas a relajar, para el cual habrá que señalar que la tolerancia para detener el proceso fue de  $1 \times 10^{-15}$  con aproximadamente un total de 9 millones de iteraciones.



Cuadro 2.4: Análisis del desplazamiento de las partículas (en escala logarítmica en el panel izquierdo) y su vector de desplazamiento (panel derecho). Aunque las partículas se siguen desplazando hacia una posición óptima, la magnitud del desplazamiento promedio de las partículas decrece rápidamente.

La imagen de la izquierda representa al desplazamiento promedio de las partículas conforme se desarrollaba el proceso. En esta se puede ver con facilidad cómo las partículas llevan una clara tendencia a moverse cada vez menos, a pesar de que el sistema aún no ha llegado a una configuración óptima. La imagen de la derecha presenta a los vectores de desplazamiento de cada una de las partículas durante la última iteración; en esta se puede notar que las partículas sí se están desplazando en la dirección aparentemente correcta y por tanto el problema más que nada recae en el hecho de por qué se desplazan tan poco.

Para encontrar el problema empecemos por notar que en las configuraciones finales de ambos casos presentados se observa un curioso patrón; en ambos existen pares de partículas muy juntas (lo cual no debería ocurrir pues se esperaría que el método las separe por sí solo) y relativamente aisladas con respecto a sus partículas vecinas; incluso en el caso de 16 partículas aleatorias, éstas se encuentran prácticamente una encima de la otra.

Lo anterior es importante porque dado que las partículas se mueven poco, sabemos que el empuje dado por el gradiente de la función error es pequeño, y eso sólo ocurrirá cuando el perfil de densidad sea numéricamente similar al perfil de densidad promediado mediante el método SPH; esto quiere decir que el método está considerando que todas aquellas partículas que se mueven poco en torno a los pares de partículas y estas mismas, tienen un buen perfil de densidad. Lo que ocurre, y que es documentado por Rosswog [14], es que el método de SPH provoca que las partículas que se acercan demasiado se comporten como una sola, empujando así a las demás que tendrían menor masa que este par de partículas pegadas. Por ello es que se encuentra un patrón en las configuraciones finales, en el cual parece haber

huecos en torno a las partículas “pegadas”.

Así pues, dado el problema de los pares, el método de relajación basado en el método del gradiente intenta relajar al sistema bajo una configuración en la cual ahora existe una partícula de masa dos veces mayor a la de las demás.

Evadir este problema con algún tipo de “pequeña” patada a las partículas pegadas no funciona, dado que las demás partículas vecinas en torno a estas dos tienden a empujarlas de vuelta, debido a que los promedios a la SPH no cambiarán mucho con respecto a los calculados para una iteración anterior. Sólo si la patada es lo suficientemente grande conseguiremos despegar las partículas, pero con ello se paga el precio de desconfigurar al sistema produciendo que se genere un nuevo proceso de relajación, en el cual se observa el mismo problema tras un cierto número de iteraciones.

En las siguientes secciones se presentarán mas casos en los que este problema se observa, así como una medida de que tan perjudicial es para la búsqueda de los datos iniciales óptimos.

# Capítulo 3

## Análisis de resultados.

### 3.1. Caso perfil de densidad constante en 2D.

Construido el código para el proceso de relajación y corroborado su funcionamiento con modelos sencillos, resta analizar su funcionamiento para diversos casos más complicados. Para este caso se emplean todos los parámetros establecidos en la sección 2.4, a la par de que el análisis se llevará a cabo para tres diferentes versiones del método de relajación.

#### 3.1.1. Método del gradiente.

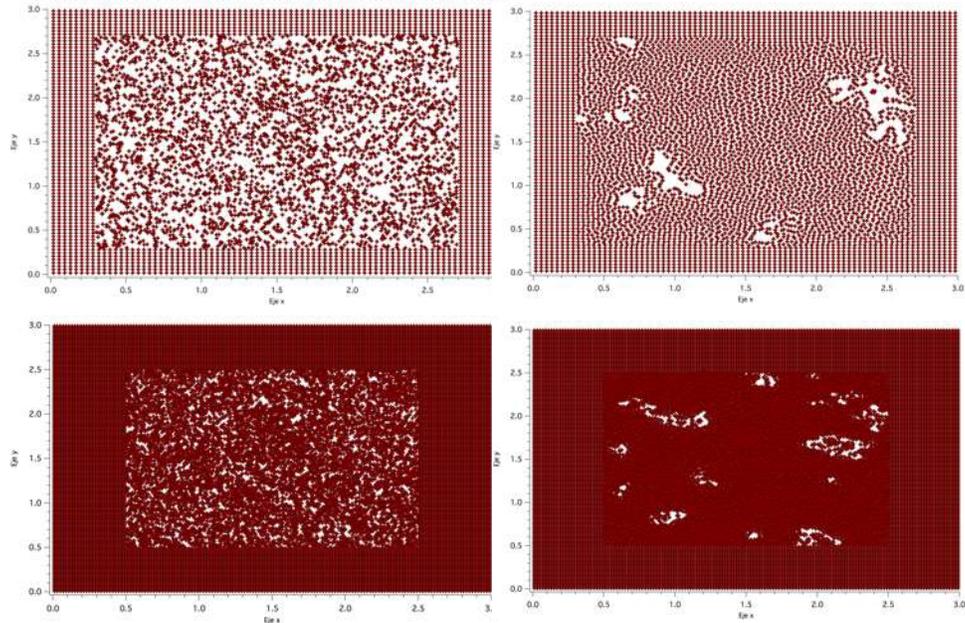
Empezando por el caso sencillo de una configuración en dos dimensiones con un perfil de densidad constante igual a 1, al aplicar el código basado en el método del gradiente podemos encontrar en el cuadro 3.1 cómo para los casos de 4096 y 10000 partículas aleatorias con  $h$  fija, el problema de los pares se mantiene presente, lo cual deja en claro que el problema se presentará en cualquier configuración de partículas y para cualquier resolución, tal como se esperaba <sup>1</sup>.

No obstante el problema de los pares, se hace presente el hecho de que conforme aumentamos el número de partículas a relajar, los agujeros formados debido a dicho problema se van reduciendo paulatinamente (comparando estos resultados con los presentados para los análisis preliminares), por lo que se puede establecer que para configuraciones masivas de partículas, éstos apenas serán perceptibles.

Por otro lado, se debe considerar también el hecho de que en todos los procesos anteriores las configuraciones aún pueden relajarse más, pero conseguir una mejoría

---

<sup>1</sup>Este problema se presenta también en las implementaciones para la versión del gradiente conjugado y de la búsqueda inexacta del paso.



Cuadro 3.1: Proceso de relajación para una configuración con densidad constante de 4096 (panel superior) y 10000 (panel inferior) partículas aleatorias en 2D mediante el método del gradiente.

significativa requiere de un extenso tiempo de procesamiento.

Por ejemplo, en el caso de 10000 partículas aleatorias a relajar, omitiendo las fronteras, estas partículas deben acomodarse en una región de  $2 \times 2$  unidades, lo cual quiere decir que las partículas idealmente deberían de conformar una rejilla de  $100 \times 100$  partículas con una separación de 0.02 unidades entre cada una de ellas. Luego, si observamos que para las últimas iteraciones realizadas en el proceso de relajación el empuje que se le da a cada una de esas partículas oscila entorno a  $8 \times 10^{-8}$ , y que dicho valor -dado que depende de la derivada de la densidad- tenderá a reducirse cada vez más conforme nos acerquemos al mínimo de la función error, podemos darnos cuenta de que cada vez será considerablemente más costoso en tiempo de procesamiento el obtener una mejor optimización para los datos iniciales.

No obstante lo anterior, sabemos que lo importante a analizar en cuanto al funcionamiento de nuestro código para la optimización de los datos iniciales, está en corroborar qué tan bueno es el perfil de densidad obtenido mediante el método del gradiente; lo cual se ejemplifica en la figura 3.1, donde se presenta el desplazamiento promedio para una configuración de 10000 partículas. El desplazamiento promedio se calcula sumando los empujes (en valor absoluto) dados por el método del gradiente a cada una de las partículas a relajar y dividiendo posteriormente el valor obtenido

entre el número total de partículas a relajar.

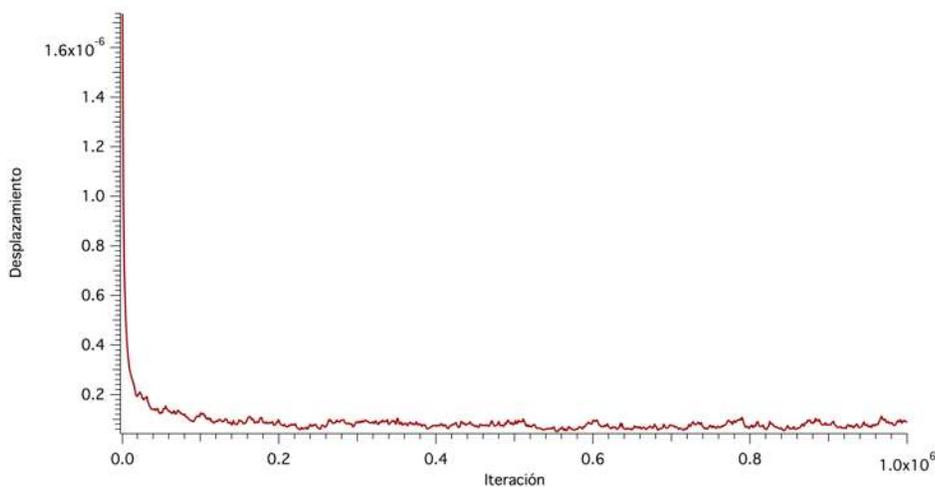
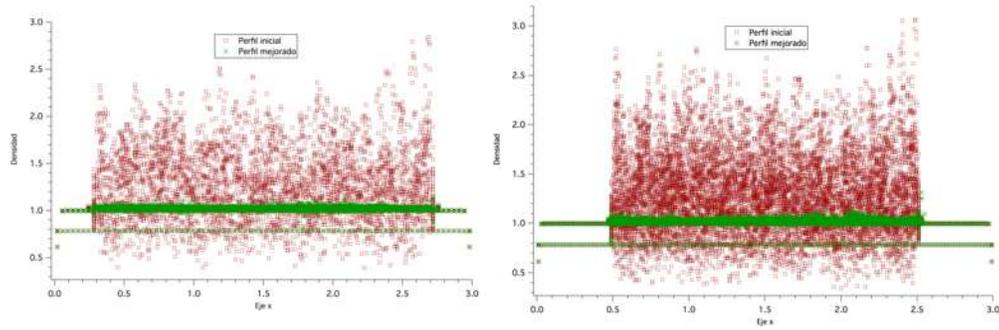


Figura 3.1: Gráfica de la magnitud promedio del desplazamiento de las partículas contra el tiempo durante el proceso de relajación de una configuración constante en 2D de 10000 partículas aleatorias.

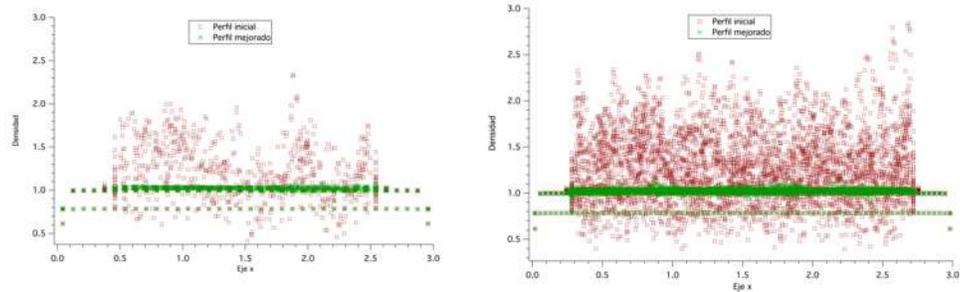
El cuadro 3.2 nos muestra los perfiles de densidad inicial y final para datos generados con el método de aceptación y rechazo, y relajados con nuestro método del gradiente para configuraciones de 4096 y 10000 partículas aleatorias. En las mismas se puede ver contundentemente la mejora en los promedios de la densidad a la SPH, lo que conlleva a que el perfil de densidad del sistema sea a su vez claramente más aproximado al perfil de densidad analítico, esto en comparación con el perfil de densidad inicial sin relajar.

### 3.1 Caso perfil de densidad constante en 2D.



Cuadro 3.2: Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 4096 (panel izquierdo) y 10000 (panel derecho) partículas a relajar empleando una longitud  $h$  fija.

A su vez, para el caso en que se tiene una  $h$  variable se encontró también una clara mejora en el perfil de densidad, tal como se muestra en el cuadro 3.3.



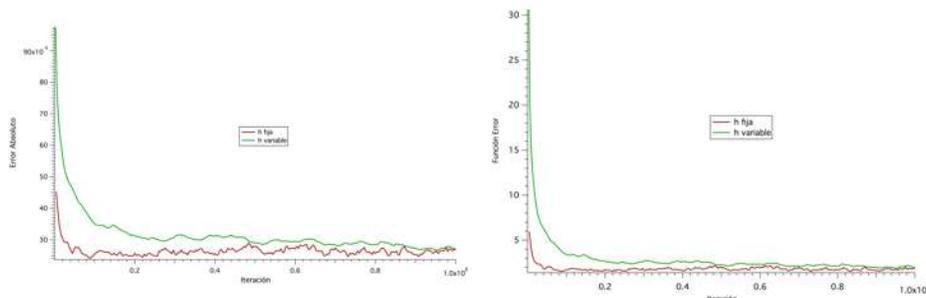
Cuadro 3.3: Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 576 (panel izquierdo) y 4096 (panel derecho) partículas aleatorias empleando una longitud  $h$  variable.

A fin de comparar estos resultados y poder poner un valor numérico en la precisión del código, recurramos al análisis del comportamiento tanto de la función error como del error absoluto promedio <sup>2</sup>. Para ejemplificar, tomemos la configuración de 4096 partículas aleatorias presentada en el cuadro 3.4 y notemos que el uso de  $h$  fija nos arroja un descenso más pronunciado hacia el 0, esto tanto para la gráfica del error absoluto como para la gráfica de la función error <sup>3</sup>. A pesar del hecho de que para la longitud  $h$  fija el descenso pronunciado en las gráficas parezca ser inicialmente mejor que el descenso suave que se tiene para la longitud  $h$  variable, habrá que recalcar que para ambas gráficas los resultados parecen tener una

<sup>2</sup>Error absoluto entre la densidad analítica y la densidad promedio a la SPH.

<sup>3</sup>Este resultado se ha encontrado también para otras versiones del método.

tendencia a ser muy parecidos conforme se deja correr el proceso de relajamiento por más tiempo.



Cuadro 3.4: Comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes  $h$  fija y  $h$  variable, en una configuración con densidad constante 2D de 4096 partículas aleatorias.

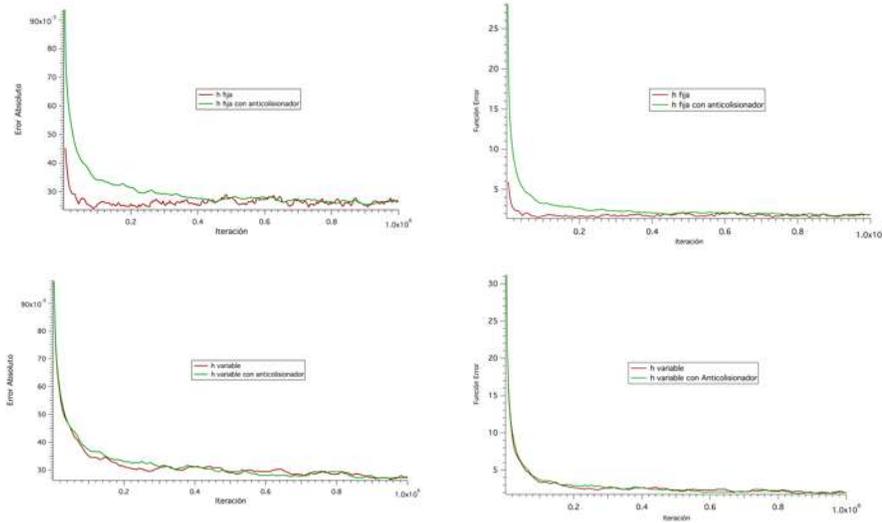
Con la intención de corregir el problema de los pares, se incorporó a nuestro método de relajación la herramienta que hemos llamado “anticolisionador”,<sup>4</sup> cuyo fin es evitar que las partículas que tienden a acercarse lleguen a hacerlo demasiado, de tal modo que se evite que afecten con ello a los promedios realizados mediante el método SPH calculados para sus partículas vecinas.

Para visualizar comparativamente el posible beneficio de esta herramienta, consideremos el cuadro 3.5, en el que para una configuración de 4096 partículas aleatorias se grafican tanto el error absoluto como la función de error, ambos casos presentándose con y sin el anticolisionador tanto para  $h$  fija como para  $h$  variable.

De lo anterior podemos ver que la incorporación del anticolisionador nos arroja un comportamiento suave conforme las gráficas se van aproximando a 0 para todos los casos. En el caso donde se emplea  $h$  variable, se nota que tanto para el error absoluto como para la función error, la gráfica con el anticolisionador muestra un comportamiento similar con respecto a la gráfica del comportamiento del proceso sin el mismo, por lo que no hay mejora en el uso del anticolisionador. Por otro lado, para la longitud  $h$  fija, el anticolisionador tampoco aporta mejora alguna salvo si buscamos un comportamiento más suave en el modo en que nos acercamos a 0.

<sup>4</sup>Ver apéndice C.

### 3.1 Caso perfil de densidad constante en 2D.



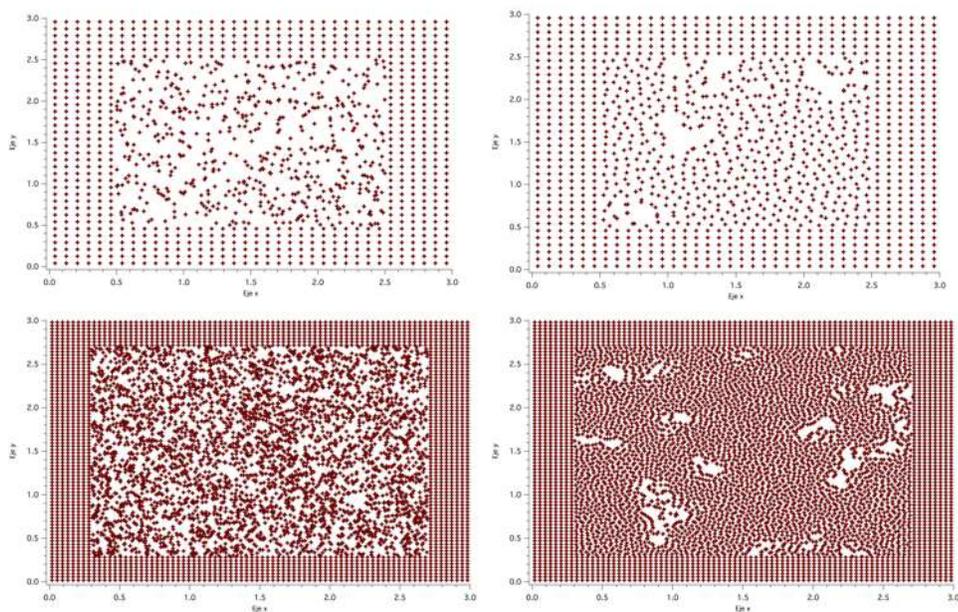
Cuadro 3.5: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos en los que se emplean longitudes  $h$  fija y  $h$  variable, en una configuración con densidad constante 2D de 4096 partículas a relajar; las gráficas inferiores representan los casos respectivos del error absoluto (a la izquierda) y la función error (a la derecha) con el uso del anticollisionador.

#### 3.1.2. Método del gradiente conjugado.

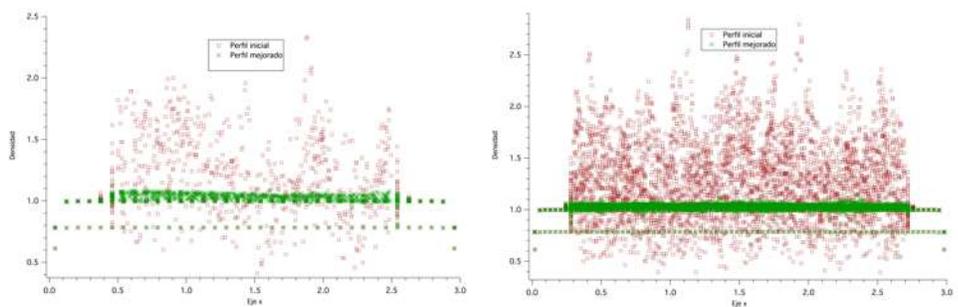
Para el caso ya mencionado de un perfil de densidad constante en 2D, se analizó también el funcionamiento de versiones alternas al método del gradiente. Concretamente, el método del gradiente conjugado y el uso de la búsqueda inexacta del paso; ambas versiones descritas en el capítulo 1. Es importante señalar que, para la versión del gradiente conjugado, el código no cambia en su construcción debido a que el modo en que se calcula la dirección de máximo cambio es idéntica para todas las versiones. La diferencia existente entre el código para el método del gradiente y la versión del gradiente conjugado, está específicamente en la forma en que se dan los empujes (pasos) a las partículas en cada una de estas versiones.<sup>5</sup>

Para esta versión se obtuvieron resultados similares a los anteriores, lo cual se puede apreciar en los cuadros 3.6 y 3.7, en los que se presentan configuraciones de 576 y 4096 partículas aleatorias, donde se ha usado la longitud  $h$  fija (tal como se hizo en algunos casos anteriores con el método del gradiente).

<sup>5</sup>Lo cual está descrito en el capítulo 1, para más detalles ver apéndice A.



Cuadro 3.6: Proceso de relajación para una configuración con densidad constante de 576 (panel superior) y 4096 (panel inferior) partículas aleatorias en 2D, mediante el método del gradiente conjugado.

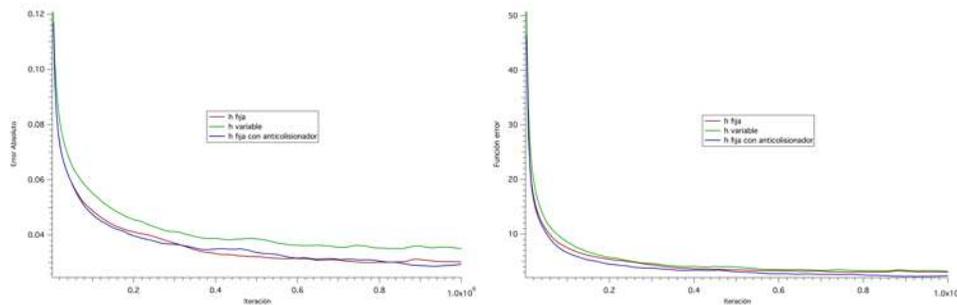


Cuadro 3.7: Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 2D, para 576 (panel izquierdo) y 4096 (panel derecho) partículas aleatorias con una longitud  $h$  fija.

En cuanto a las gráficas para el error absoluto (ver cuadro 3.8), si comparamos los casos para longitud  $h$  variable, longitud  $h$  fija y la implementación del anticollisionador, notaremos que de nuevo para la  $h$  variable se tiene un peor resultado final en relación a cuando empleamos la  $h$  fija; esto es importante porque con la  $h$  fija se requiere sólo de cerca de  $3/4$  del tiempo de procesamiento que se necesita para un mismo caso con  $h$  variable; por otro lado, incorporar el anticollisionador para

intentar mejorar el resultado del caso con la  $h$  fija no muestra mejora significativa.

En la gráfica de la función error (ver cuadro 3.8), se nota también el hecho de que el uso de  $h$  variable no es la opción mas conveniente para llevar a cabo el proceso de relajación; lo cual nos permite ir dejando de lado esta idea si nuestra intención es encontrar un acomodo cada vez más óptimo para las partículas.



Cuadro 3.8: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes  $h$  fija,  $h$  variable y  $h$  fija con anticollisionador, para una configuración con densidad constante 2D de 4096 partículas aleatorias.

### 3.1.3. Búsqueda lineal inexacta del paso.

La última versión del código a analizar es el uso de la búsqueda inexacta del paso <sup>6</sup>. Realizando, al igual que en las versiones anteriores, un análisis para la  $h$  fija, la  $h$  variable y el anticollisionador, se encontró una buena mejora para el perfil de densidad con respecto al perfil de densidad de los datos iniciales, tal como se muestra en la figura 3.2 (dado el alto coste en tiempo de programación que se requirió para la implementación de esta versión del método, fue necesario realizar los procesos a analizar con un número menor de partículas).

<sup>6</sup>Para más detalles ir al capítulo 1.

### 3.1 Caso perfil de densidad constante en 2D.

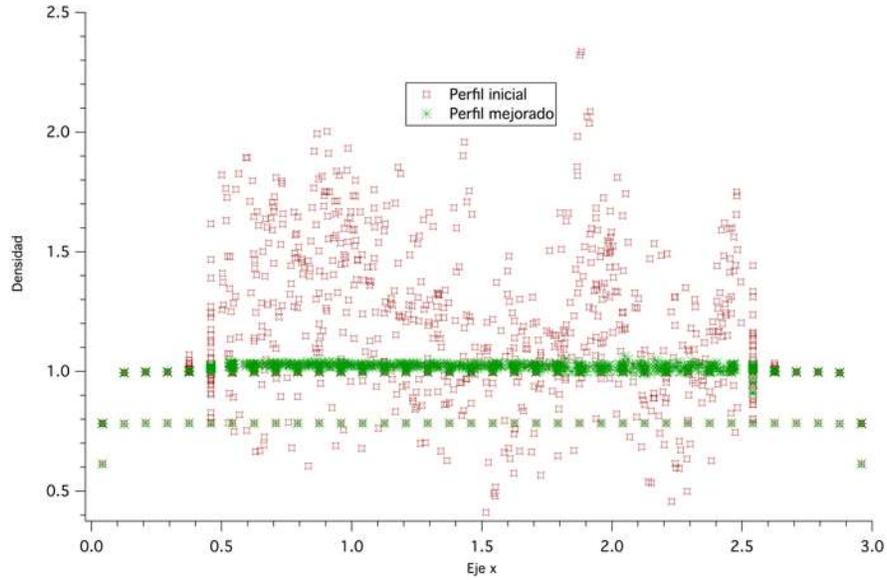
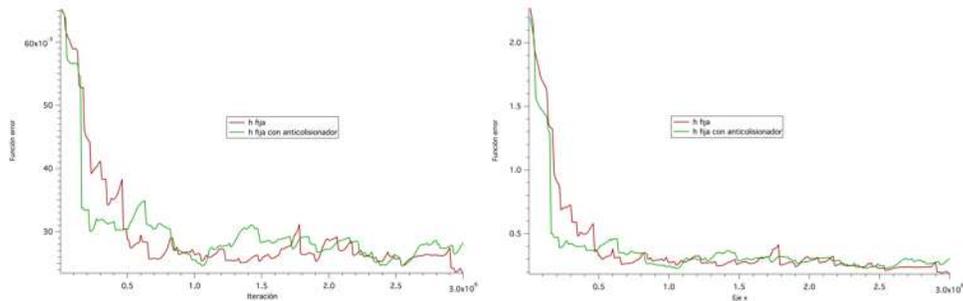


Figura 3.2: Análisis del perfil de densidad final e inicial para una configuración con densidad constante en 2D, para 576 partículas aleatorias con una longitud  $h$  fija.

No obstante, cuando analizamos las gráficas del error absoluto y la función error (presentadas en el cuadro 3.9), encontramos un comportamiento irregular en comparación con lo antes presentado. Omitiendo el caso para la longitud  $h$  variable, que de nuevo nos arroja el peor resultado, podemos ver que en la gráfica de  $h$  fija *vs*  $h$  fija con anticollisionador, no hay una diferencia marcada entre ninguno de los casos.



Cuadro 3.9: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con una longitud  $h$  fija y  $h$  variable, para una configuración con densidad constante 2D de 576 partículas aleatorias.

Si bien el método de la búsqueda lineal inexacta nos permite hacer un proceso de relajación con resultados favorables, habrá que tomar en cuenta el hecho de que el uso de la búsqueda inexacta requiere un tiempo de procesamiento siete veces superior a su contra parte para las versiones con el método del gradiente o el gradiente conjugado (cuyos tiempos de procesamiento son prácticamente similares entre si).

## 3.2. Caso perfil de densidad constante en 3D.

El paso inmediato después de lo descrito en la sección anterior es analizar ahora el caso de una densidad constante en una configuración 3 dimensional, tal como si estuviésemos conformando un cubo. Estableciendo los mismos parámetros empleados hasta ahora (fronteras fijas, valor para  $\eta$ , etc.), sólo será necesario ajustar el método del gradiente para su correcto funcionamiento con  $N$  partículas en 3 dimensiones y ajustar a su vez el cálculo de los promedios a la SPH <sup>7</sup>.

Dado que para un caso tridimensional, la longitud  $h$  nos permitirá calcular promedios a la SPH para una partícula  $i$ -ésima mediante las partículas  $j$ -ésimas vecinas, contenidas en una esfera conformada por esa  $h$  (de radio  $2h$  para el Kernel  $CS$ ) y que en dos dimensiones dichas partículas vecinas estarán contenidas en un círculo (también de radio  $2h$ ), debe quedar claro que cuando el método del gradiente realice el proceso de relajación requerirá mucho más tiempo de cómputo para calcular cada uno de los promedios necesarios.

Debido a su alto costo de procesamiento y a que no aporta considerable mejoría en la optimización de la posición de las partículas, se han dejado de lado los procesos que involucran la búsqueda inexacta del paso. Estos resultados serán retomados en las conclusiones cuando se hable de la ventaja y desventaja de usar uno u otro método.

### 3.2.1. Método del gradiente.

Al igual que para el caso en 2D, el método del gradiente en 3D nos permite alcanzar una aceptable mejoría con respecto a las posiciones iniciales de las partículas generadas mediante el método de aceptación y rechazo. Esto se puede ver en la figura 3.3 donde presentamos el perfil de densidad inicial contra el perfil de densidad

---

<sup>7</sup>Sólo es necesario modificar el Kernel, el cálculo de la longitud  $h$  y establecer las condiciones iniciales para una configuración en 3 dimensiones.

optimizado mediante el método del gradiente, esto para una configuración de 13824 partículas aleatorias.

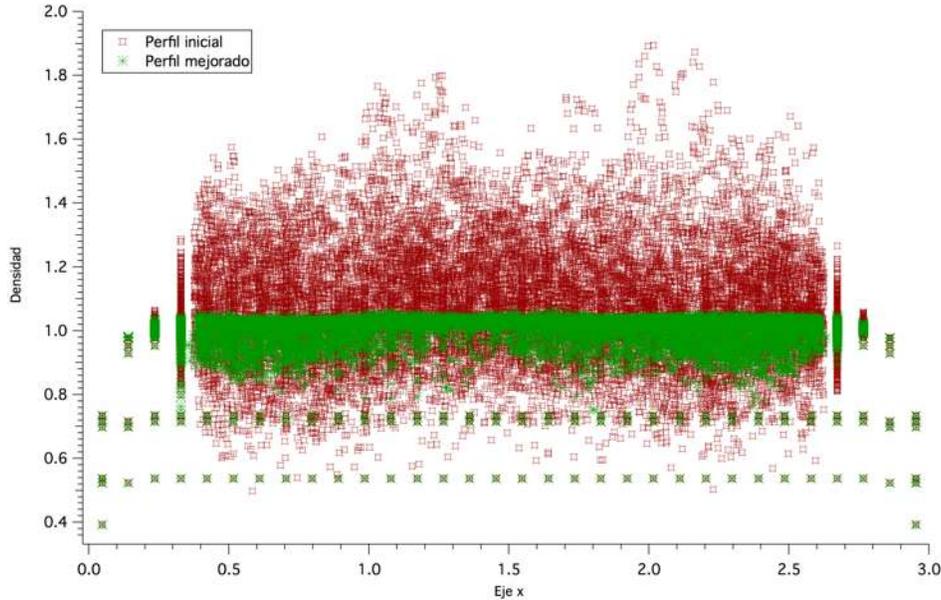


Figura 3.3: Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 3D para 13824 partículas aleatorias con una longitud  $h$  fija.

Recordemos que los promedios a la SPH se ven afectados en las fronteras por la falta de vecinos para calcular correctamente dichos promedios y que por ello es que hemos generado una rejilla de partículas frontera fijas, las cuales tienen una posición correcta que no será modificada por el método del gradiente.

Para establecer un número de iteraciones a partir de las cuales podamos realizar una comparación con otros casos y métodos, es necesario monitorear el desplazamiento de las partículas; esto a fin de estimar un criterio de parada similar para todos los procesos a comparar. Tal como se ve en la figura 3.4, donde se emplea escala logarítmica, el desplazamiento de las partículas decae rápidamente conforme el proceso de relajación avanza, hasta que en cierto punto se llega a un lapso en el que el desplazamiento promedio tiene un comportamiento ligeramente irregular, pero con una marcada tendencia a decrecer paulatinamente.

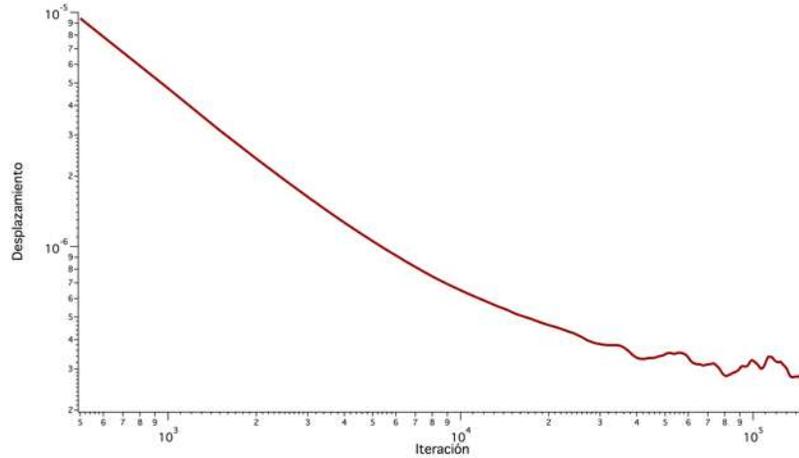
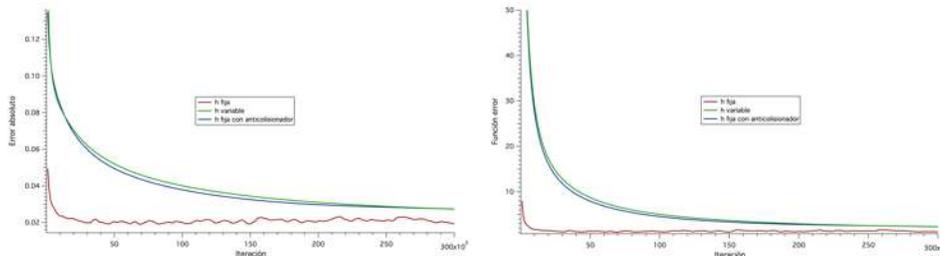


Figura 3.4: Gráfica en escala logarítmica del desplazamiento promedio de las partículas durante el proceso de relajación.

Continuando con el análisis del método en 3D al recaer en el error absoluto y la función error (ver cuadro 3.10), nos encontramos con un comportamiento prácticamente idéntico al observado para el caso en 2D. Es decir, para  $h$  fija el método cae abruptamente hasta empezar a detenerse súbitamente y a partir de entonces decrece muy lentamente, mientras que al aplicar el anticollisionador con  $h$  fija encontramos un descenso sumamente suave que, después de un largo tiempo de procesamiento, va acercándose a los valores obtenidos por  $h$  fija sin el anticollisionador.



Cuadro 3.10: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con una longitud  $h$  fija sin anticollisionador y una  $h$  fija con anticollisionador para una configuración con densidad constante 3D de 4096 partículas aleatorias.

### 3.2.2. Método del gradiente conjugado.

Para el método del gradiente conjugado se obtuvieron también nuevamente resultados aceptables a la hora de mejorar nuestro perfil de densidad inicial. En la

figura 3.5 se muestra justamente esa mejoría con respecto a un perfil inicial de 13824 partículas, e incluso, si comparamos este resultado con lo presentado para el método del gradiente en 3 dimensiones para un mismo número de partículas, se puede notar que con el método del gradiente conjugado se obtiene un perfil de densidad mejorado más compacto <sup>8</sup>.

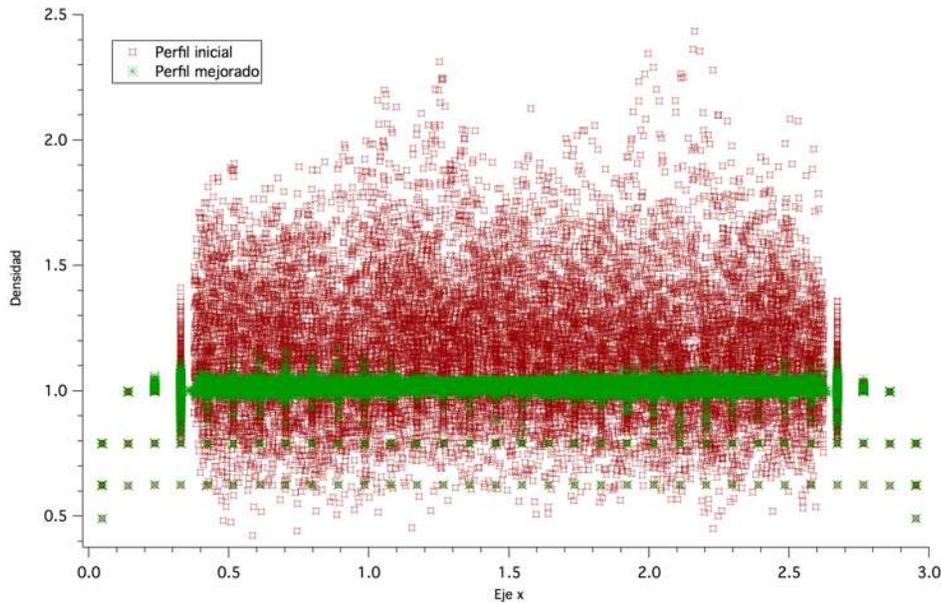
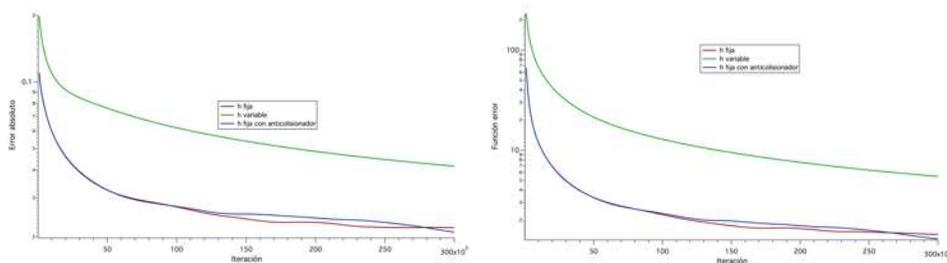


Figura 3.5: Análisis del perfil de densidad final e inicial para un configuración con densidad constante en 3D para 13824 partículas aleatorias con una longitud  $h$  fija.

A la par de lo anterior, comparando con el método conjugado en 2D, encontramos de nuevo que para el error absoluto y la función error (ver cuadro 3.11), las gráficas para los procesos con  $h$  fija y  $h$  fija con anticollisionador no difieren mucho e incluso en gran parte del proceso de relajación se mantienen muy cercanos. Sólo para corroborar los resultados con la  $h$  variable, se ha incluido la gráfica de su comportamiento, encontrando de nuevo que usar  $h$  fija es sumamente mejor tanto en precisión como en tiempo de procesamiento.

<sup>8</sup>Lo cual no quiere decir que las posiciones finales para las partículas representen de mejor manera al perfil de densidad analítico.



Cuadro 3.11: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes  $h$  fija sin anticollisionador y  $h$  fija con anticollisionador para una configuración con densidad constante 3D de 4096 partículas aleatorias.

### 3.3. Caso perfil de densidad dependiente de $r$ en 3D.

El último caso a analizar es el que denominamos en un inicio de este trabajo como nuestro caballo de batalla y que está dado por la ecuación 2.1. Esto no sólo por que la construcción del código se enfocó en cierta medida en poder optimizar una densidad del tipo  $1/r$ , sino también por que es este caso el que pondrá de manifiesto la capacidad del método para optimizar un sistema modelado por partículas (elementos de un fluido bajo el cuadro del método SPH) en el cual se tiene un perfil de densidad dependiente de la posición.

Con respecto a los casos anteriores en el que la densidad analítica era constante, ahora nuestro método trabajará con una densidad “no” constante. No obstante, ello no supone ninguna modificación al código, salvo por el hecho de ingresar al código la correcta función de densidad que especifique al sistema que se desea modelar.

En lo que respecta al buscador eficiente de vecinos <sup>9</sup>, no es necesario tampoco realizar cambio alguno pues su construcción está planteada de forma general. Los promedios realizados con el método SPH tampoco requieren ser modificados pues el kernel es el mismo que se ha empleado para los otros casos y las cuentas solo dependerán del perfil de densidad analítico que se desee modelar.

Por lo anterior, para el caso de una densidad dependiente del radio en 3 dimensiones y empleando los mismos parámetros iniciales que se han estado usando a lo largo de este trabajo, se analizan los métodos del gradiente y del gradiente conjugado considerando para cada uno de ellos el uso  $h$  fija y  $h$  variable.

<sup>9</sup>Ver apéndice B.

Hay que señalar que en el caso de la  $h$  fija, ahora realmente no tomaremos una densidad constante, sino que tomaremos una densidad dependiente de la posición por lo que el valor de la longitud  $h$  será diferente para cada partícula e incluso cambiará en cada iteración. Hay que tener en claro entonces que para la longitud  $h$ , la diferencia entre las versiones que hemos llamado “fija” y “variable” está intrínsecamente en la forma en que se introduce el valor de la densidad  $\rho$  en la ecuación

$$h_i = \eta \left( \frac{m}{\rho(\mathbf{r})_i} \right)^{\frac{1}{3}}. \quad (3.1)$$

Así, para la  $h$  fija,  $\rho$  se introduce como una densidad analítica evaluada en la posición actual de la partícula  $i$ -ésima. Por su parte, para el caso con longitud  $h$  variable,  $\rho$  se introduce como un promedio de la densidad calculada con el método SPH sobre la posición de la partícula  $i$ -ésima con respecto a la posición de sus  $j$ -ésimas partículas vecinas.

### 3.3.1. Fronteras aleatorias fijas.

Algo que debemos tener claro es que al trabajar con una densidad que depende del radio, nuestra representación del sistema en coordenadas cartesianas tendrá la forma presentada en la figura 2.1, y por tanto no es posible emplear como frontera a partículas con posición fija y con una densidad cuyo valor sea correcto en lo que respecta al perfil de densidad analítico a modelar. Esto es, para poder llevar a cabo la implementación de fronteras fijas, tendríamos que saber como colocar a las partículas en su posición de forma óptima, de tal modo que dichas posiciones reflejen un perfil analítico idéntico al perfil analítico que intentamos modelar. Para el caso de una densidad constante igual a 1, esto era fácil pues el trabajo consistía en simplemente construir una cuadrícula con los vértices bien marcados.

Necesitamos entonces, una forma de construir una región fronteriza que sirva como condición de frontera para nuestro sistema y que ayude en la optimización de los datos iniciales de forma similar a como lo hacen las fronteras que se han usado para los casos anteriores. Para hacer esto, de entre diversas posibilidades, se ha optado por emplear fronteras aleatorias fijas, más que nada por la simplicidad que en su construcción obtenemos al adaptarlas al código.

Las fronteras aleatorias fijas consisten en partículas fronterizas generadas mediante el método de aceptación y rechazo bajo el mismo criterio (mismo perfil de densidad) con el que se generan las partículas que constituyen los datos iniciales a optimizar. Lo que haremos entonces será establecer un radio máximo igual 1.5 para el proceso de generación de datos iniciales; durante el mismo, se tomarán todas las

partículas contenidas dentro de un radio igual a 1 como los datos iniciales a optimizar, y así, todas las demás partículas que estarán entre 1 y 1.5 serán tomadas como partículas fronterizas, para las cuales su posición no será modificada. Lo anterior, implica que bajo la forma en que ahora estamos generando los datos, no nos es posible fijar el número exacto para estos datos iniciales (aunque si un valor aproximado al deseado); esto dado que, para que la distribución de probabilidad que empleamos en el método de aceptación y rechazo sea consistente con el perfil de densidad analítico, es incorrecto generar por separado a las partículas que representen a los datos iniciales y a las que representen a la frontera.

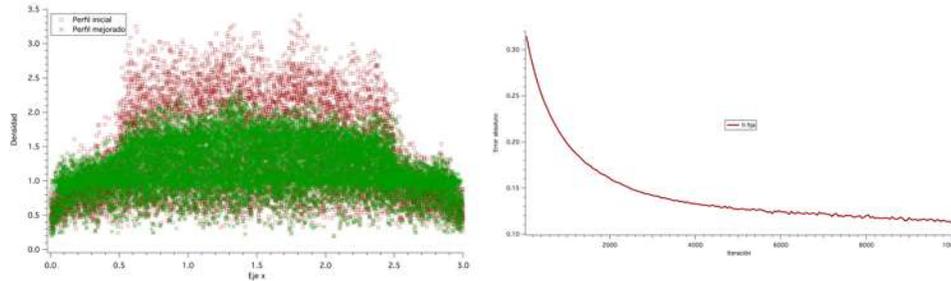
La elección del ancho de la región de fronteriza es hasta cierto punto arbitrario, lo único que se debe tener en cuenta para fijarlo, es que dicha región debe tener una longitud superior a dos veces el valor promedio de la longitud  $h$  a fin de que los promedios calculados a la SPH no queden truncados y den por tanto valores incorrectos (recordemos que para el Kernel  $CS$  el radio de influencia de las partículas vecinas es de  $2h$ ). En la practica, se encontró que para los casos en los se usaba  $h$  variable, era mejor emplear una magnitud de más de tres veces el valor promedio de dicha longitud  $h$  dado que para este caso la  $h$  puede crecer mucho cuando no tiene suficientes partículas vecinas cercanas. Así, por ejemplo, para el caso que se presentará mas adelante con 3100 partículas aleatorias a optimizar, el valor de la longitud  $h$  promedio oscilaba en torno a 0.15, por lo que un radio máximo de 1.5 mostró un buen funcionamiento tanto para  $h$  fija como para  $h$  variable.

Si bien hasta ahora se ha encontrado una forma sencilla de construir una región fronteriza, debemos tener en cuenta que ésta se halla construida con un método basado en Monte Carlo. Por tanto para corroborar que el uso de una frontera de partículas aleatorias con posición fija permite un buen funcionamiento del proceso de optimización para los datos iniciales, recurramos al caso de la densidad constante en 3D con 4096 partículas aleatorias a optimizar <sup>10</sup> y en vez de agregar al sistema una frontera de partículas bien posicionadas, agreguemos una frontera de partículas con posición aleatoria.

El cuadro 3.12 nos muestra como para sólo 10000 iteraciones ya se ve una tendencia clara de optimización de los datos iniciales. Así, en la gráfica de la izquierda nos percatamos de que el perfil de densidad promedio para los datos iniciales ha mejorado considerablemente con respecto a los valores obtenidos originalmente. Del mismo modo, para la gráfica de la derecha encontramos que el error absoluto promedio se va reduciendo de forma creciente y con una tendencia similar a los resultados anteriormente analizados.

---

<sup>10</sup>Ver sección 3.2.



Cuadro 3.12: Análisis del proceso de optimización empleando fronteras aleatorias para un caso con 4096 partículas a relajar para una densidad constante. Se observa que el perfil de densidad (panel izquierdo) obtiene una mejora considerable con sólo 10000 iteraciones. La gráfica del error absoluto (panel derecho) refleja la tendencia en la reducción del mismo.

### 3.3.2. Método del gradiente.

Trabajando con la versión del código que emplea el método de máximo descenso (método del gradiente), se hicieron pruebas para  $h$  fija y  $h$  variable, a su vez para ambos casos se realizaron pruebas similares empleando el anticollisionador. Para todos los casos, se obtuvieron mejoras aceptables para el perfil de densidad de los datos iniciales. Por ejemplo, en la figura 3.6 se compara el perfil de densidad inicial contra el perfil de densidad optimizado, este último, obtenido empleando una longitud  $h$  fija sin anticollisionador. Se puede notar que al comparar las gráficas del perfil de densidad mejorado y del perfil de densidad analítico, las curvas están prácticamente encima.

A su vez, mientras más nos acercamos a la frontera (radio igual a uno), las partículas presentan un valor promedio de la densidad que se va alejando del valor correcto. Esto se debe a que las partículas a optimizar cercanas a la frontera, toman partículas fijas de la frontera como parte de sus partículas vecinas con las que realizarán los promedios a la SPH pertinentes.

Con resultados similares a los obtenidos en las secciones 3.1 y 3.3, se encontró también que tanto para el valor del error absoluto como el de la función error, la versión del código que emplea a la longitud  $h$  fija fue la que obtuvo los mejores resultados tal como se muestra en el cuadro 3.14.

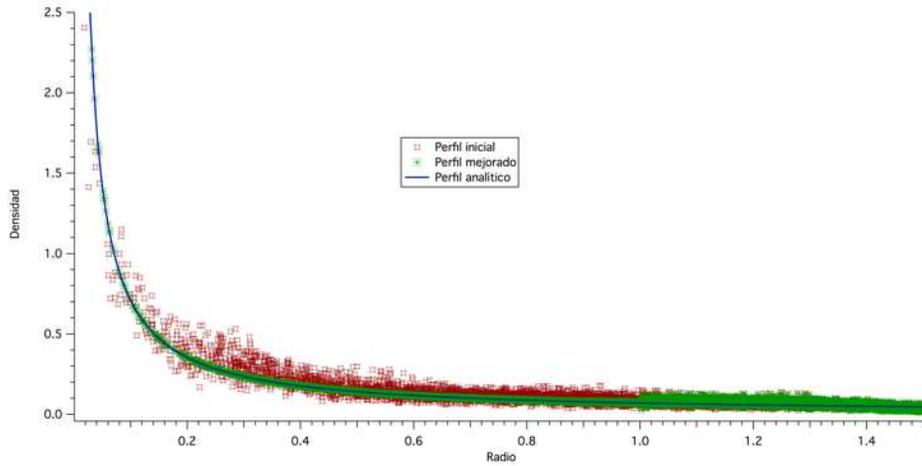
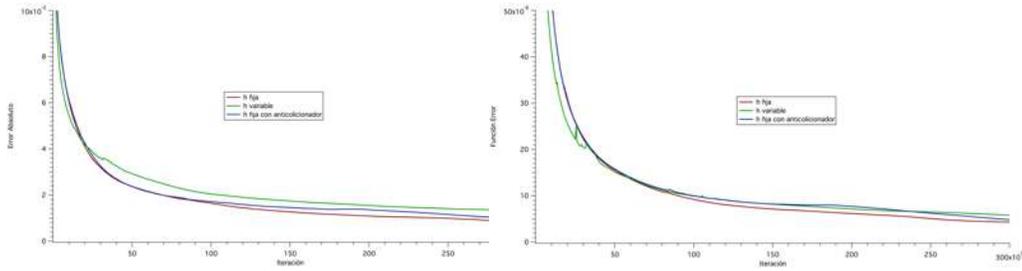


Figura 3.6: Análisis del perfil de densidad final e inicial para un configuración dependiente del radio en 3D de 3089 partículas aleatorias, con una longitud  $h$  fija y fronteras conformadas por partículas aleatorias.



Cuadro 3.13: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes  $h$  fija sin anticollisionador,  $h$  fija con anticollisionador y  $h$  variable para una configuración dependiente del radio en 3D de 3089 partículas aleatorias, empleando el método del gradiente conjugado.

### 3.3.3. Método del gradiente conjugado.

Finalmente, tal como podría esperarse, para los casos en que se empleó el método del gradiente conjugado, la tendencia hasta ahora obtenida para los resultados se mantuvo sin cambios. En la figura 3.7 se muestra, al igual que en el caso anterior, una clara mejora para el perfil de densidad tras el proceso de optimización; esta mejora fue obtenida empleando una longitud  $h$  fija sin anticollisionador. Se puede notar que al igual que para el caso empleando el método del gradiente, al comparar las gráficas del perfil de densidad mejorado y del perfil de densidad analítico, las curvas están también prácticamente encima.

### 3.3 Caso perfil de densidad dependiente de $r$ en 3D.

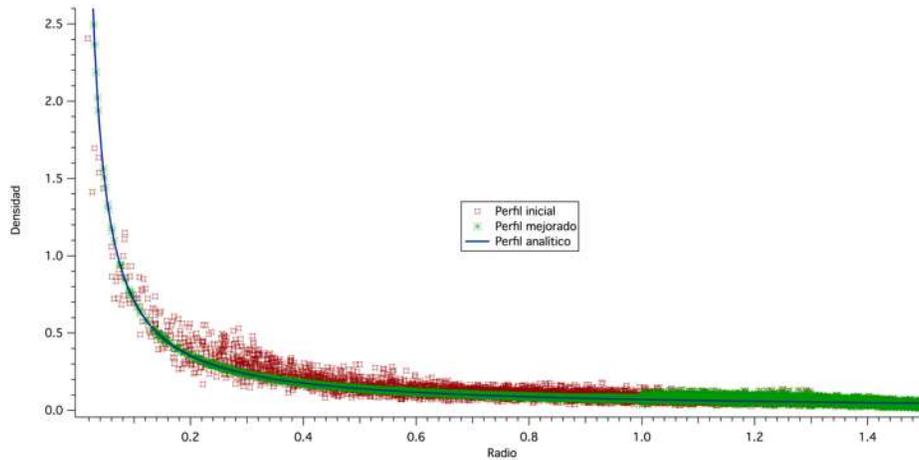
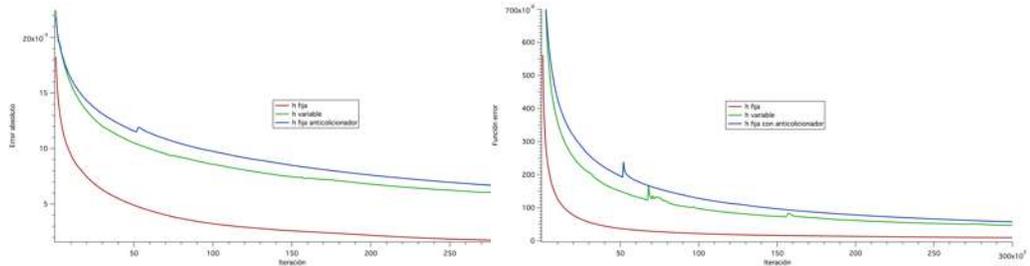


Figura 3.7: Análisis del perfil de densidad final e inicial para un configuración dependiente del radio en 3D de 3089 partículas aleatorias, con una longitud  $h$  fija.

Finalmente, el cuadro 3.14 presenta los resultados obtenidos para las gráficas del error absoluto y la función error para este caso. De nuevo se encuentra que el empleo de la longitud  $h$  fija genera los mejores resultados para el proceso de optimización.



Cuadro 3.14: Gráfica comparativa del error absoluto (panel izquierdo) y de la función error (panel derecho) para los casos con longitudes  $h$  fija sin anticollisionador,  $h$  fija con anticollisionador y  $h$  variable para una configuración dependiente del radio en 3D de 3089 partículas aleatorias, empleando el método del gradiente conjugado.

### 3.3 Caso perfil de densidad dependiente de $r$ en 3D.

---

# Capítulo 4

## Conclusiones.

A lo largo de este trabajo se ha analizado el funcionamiento de distintas versiones del método del gradiente, con el fin de adaptarlo funcionalmente como un proceso para mejorar el perfil de densidad de datos iniciales generados mediante el método de aceptación y rechazo.

Así hasta ahora se ha corroborado que emplear  $h$  fija en cualquiera de las versiones del método del gradiente es siempre superior a emplear  $h$  variable, esto tanto en la optimización como en el tiempo de procesamiento.

El problema de las partículas que se unen comportándose como una sola partícula, generando con ello malos promedios para sus vecinos próximos, es quizá el principal obstáculo en el proceso de optimización pues impide a muchas de las partículas el poder llegar a una posición adecuada, arrastrando con ello un error en las posiciones que se extiende a lo largo de la configuración de partículas.

Se encontró también que el anticollisionador implementado justamente como un intento para amortiguar el efecto de este problema no tuvo el resultado esperado, pues para ninguna de las versiones proporcionó una mejora notoria en el perfil de densidad mejorado.

Finalmente es importante señalar cuál de las versiones del método del gradiente analizadas en este trabajo proporcionó los mejores resultados para el proceso de relajación de las partículas. Para ello se presenta un esquema comparativo de las tres versiones del método para una configuración idéntica de 4096 partículas en 2D (ver figura 4.1), en ésta queda de manifiesto la relativa superioridad del método del gradiente con respecto a las otras dos versiones del mismo.

En este respecto habría que señalar que el descenso abrupto del método del

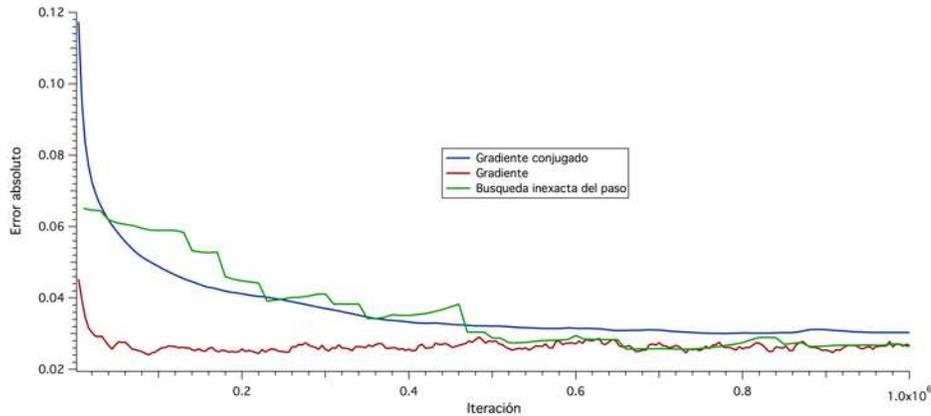


Figura 4.1: Gráfica comparativa del error absoluto para los métodos del gradiente, búsqueda inexacta del paso y del gradiente conjugado para un configuración de 4096 partículas en 2D.

gradiente durante las primeras iteraciones deja muy atrás a los otros métodos en cuanto a la reducción del error absoluto y el valor de la función error, y de hecho alcanza siempre con rapidez una optimización para las posiciones muy similar a la que obtendría si se le deja correr por un tiempo considerablemente largo; con ello, el método del gradiente otorga una optimización aceptable y rápida para los datos iniciales.

No obstante tal como se muestra en la gráfica 4.1, la búsqueda inexacta del paso llegará a un resultado similar al método del gradiente si se le da el tiempo necesario, no obstante para un mismo número de iteraciones el método del gradiente con búsqueda inexacta del paso siempre requerirá un tiempo de procesamiento por lo menos siete veces mayor. Luego, si consideramos que el método del gradiente alcanza un valor de optimización aceptable en poco tiempo, entonces emplear la búsqueda inexacta del paso no otorga ventaja alguna.

Caso diferente es el método del gradiente conjugado, del cual habría que decir, requiere un tiempo de procesamiento prácticamente idéntico al del método de gradiente ordinario. A la par de ello, el método del gradiente conjugado obtuvo mejores resultados para las configuraciones en 3D, tal como se muestra en la figura 4.2 en la cual se ejemplifica el caso con una configuración de 4096 partículas en 3D.

Aunque a su vez, los resultados obtenidos para el caso de una densidad dependiente del radio (ver figura 4.3), muestran al método del gradiente con un valor para el error absoluto algo alejado y por debajo del valor obtenido para el método

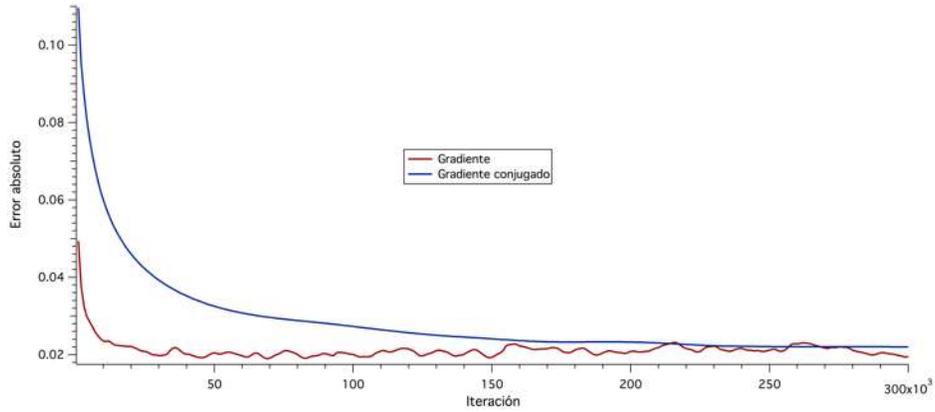


Figura 4.2: Gráfica comparativa del error absoluto para los métodos del gradiente y del gradiente conjugado para un configuración de 4096 partículas en 3D.

del gradiente conjugado. El hecho de no tener una constancia en este hecho se debe a la necesidad realizar pruebas con un considerable mayor número de iteraciones.

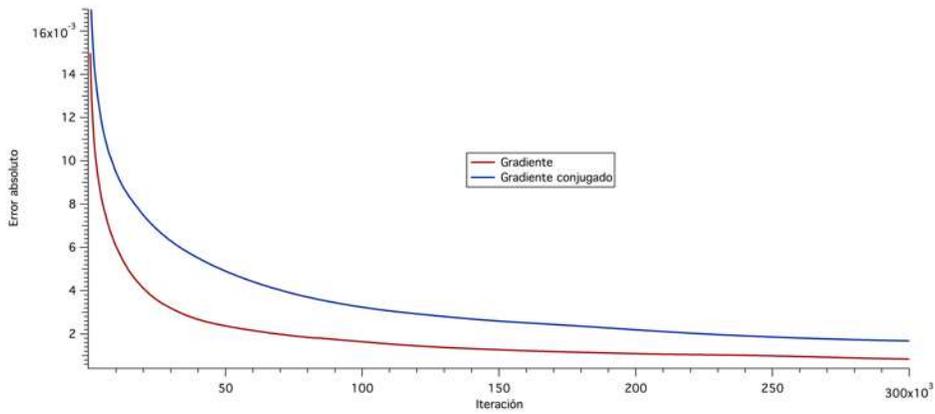


Figura 4.3: Gráfica comparativa del error absoluto para los métodos del gradiente y del gradiente conjugado para un configuración de 4096 partículas en 3D.

De hecho, siendo rigurosos, sería necesario dejar correr los procesos hasta que estos hayan alcanzado un cero computacional en el desplazamiento de las partículas para poder afirmar que versiones del código otorgan los mejores resultados; no obstante, el largo tiempo de procesamiento y el carácter intrínseco del método del gradiente (y las versiones derivadas del mismo) en el cual, conforme la función a minimizar se acerca a 0 genera pasos cada vez mas cortos, hicieron poco viables

estas pruebas al considerar el poco tiempo que se tenía disponible para concluir este trabajo.

Debe recalcar también, dado lo anterior, que para casos en los que se desee obtener una optimización muy profunda para las posiciones de los datos iniciales <sup>1</sup>, el método del gradiente y cualquiera de sus versiones aquí probadas no serían la mejor opción por el problema de las partículas que se unen y por el tiempo de procesamiento.

De todo lo anterior, se puede concluir que la adaptación del método del gradiente tiene éxito al ser empleado para optimizar las posiciones de partículas (elementos de un fluido) que fungen como datos iniciales que representan a un sistema físico; donde dichas posiciones son generadas mediante el método de aceptación y rechazo, y las propiedades físicas del sistema, como la densidad del sistema o la derivada del mismo, son calculadas mediante promedios empleando el método SPH.

## 4.1. Perspectivas.

Como perspectivas futuras para este trabajo se tiene lo siguiente:

1. Realizar pruebas al código con procesos que se detengan hasta un límite computacional, esto es, hasta que el desplazamiento de las partículas sea cercano a ser nulo computacionalmente.
2. Realizar pruebas para otras funciones dependientes de la posición que presenten otras características interesantes o que puedan presentar mayor complejidad en el procesamiento del código.
3. Adaptar al código a otras versiones del método del gradiente mas populares como es el caso del método espectral del gradiente [15] [16].

---

<sup>1</sup>Esto es, por ejemplo un error absoluto final rondando valores de  $1 \times 10^{-13}$ .

# Apéndice A

## Método de las direcciones conjugadas.

La experiencia ha demostrado que las direcciones llamadas conjugadas son mucho más efectivas como direcciones de búsqueda que otras, como pueden ser las direcciones ortogonales (características del método de máximo descenso).

Dos direcciones  $\mathbf{S}^i$  y  $\mathbf{S}^j$  se dice que son *conjugadas* una con respecto a la otra si:

$$(\mathbf{S}^i)^T \mathbf{Q} (\mathbf{S}^j) = 0. \quad (\text{A.1})$$

En general, un conjunto de  $n$  direcciones de búsqueda linealmente independientes  $\mathbf{S}^0, \mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^{n-1}$  se dice que son conjugadas con respecto a una matriz definida positiva  $Q$  si  $(\mathbf{S}^i)^T \mathbf{Q} (\mathbf{S}^j) = 0$ , para  $1 \leq i \neq j \leq n$ .

Cuando realizamos un proceso de optimización, la matriz  $Q$  es la matriz hessiana  $H$  de la función objetivo. Para una función cuadrática  $f(\mathbf{x})$  de  $n$  variables, para la cual  $H$  es una matriz constante, está garantizado que se obtiene el óptimo en  $n$  etapas de búsqueda unidireccional si se obtiene exactamente el mínimo de cada etapa.

La ortogonalidad es un caso especial de la conjugación cuando  $\mathbf{Q} = \mathbf{I}$ . Aunque es común encontrar en la bibliografía conjuntos de métodos conocidos como de direcciones conjugadas, estas, estrictamente hablando, sólo existen para funciones cuadráticas o aproximaciones cuadráticas de la función objetivo en la etapa  $k$ .

¿Cómo se pueden calcular las direcciones conjugadas sin usar derivadas? Este es un concepto sencillo al que haremos referencia en la figura A.1.

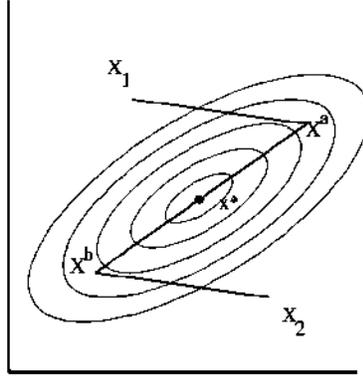


Figura A.1: Direcciones Conjugadas.

Comenzamos con el punto  $\mathbf{x}^0$ . Localizamos el punto  $\mathbf{x}^a$  que es el mínimo en una dirección cualquiera  $\mathbf{S}$ . Elegimos otro punto cualquiera (distinto del primero)  $\mathbf{x}^1$ , y localizamos el punto  $\mathbf{x}^b$  que es el mínimo partiendo del punto  $\mathbf{x}^1$  en la misma dirección  $\mathbf{S}$ . Los puntos  $\mathbf{x}^a$  o  $\mathbf{x}^b$  se obtienen minimizando  $f(\mathbf{x}^0 + \alpha\mathbf{S})$  con respecto a  $\alpha$ , admitiendo que  $f(\mathbf{x})$  es una función cuadrática:

$$f(\mathbf{x}) = f(\mathbf{x}^0) + \nabla^T f(\mathbf{x}^0) \Delta\mathbf{x}^0 + \frac{1}{2} (\Delta\mathbf{x}^0)^T \mathbf{H} (\Delta\mathbf{x}^0). \quad (\text{A.2})$$

Se puede demostrar que el óptimo de una función cuadrática cae en la línea que une  $\mathbf{x}^a$  con  $\mathbf{x}^b$ . Sin embargo, esta última afirmación no es válida para otro tipo de funciones.

*Powell* [17] desarrolló un método basado en las direcciones conjugadas antes descritas y que es aplicable a cualquier tipo de funciones; aunque *Powell* introdujo importantes modificaciones a su método para conseguir la adaptación. El método sigue una serie de pasos descritos a continuación (ver figura A.2 para el caso en dos dimensiones).

Paso 1.- El método comienza realizando una búsqueda en  $n$  direcciones linealmente independientes  $(\mathbf{s}_1^0, \mathbf{s}_2^0, \dots, \mathbf{s}_n^0)$  que suelen tomarse paralelas a los ejes coordenados. Partiendo del punto base  $\mathbf{x}_0^0$ , se lleva a cabo una búsqueda unidireccional en la dirección  $\mathbf{s}_1^0$  para llegar al punto  $\mathbf{x}_1^0$ . Este punto  $(\mathbf{x}_1^0)$  se toma como punto de partida para una nueva búsqueda unidireccional, en este caso en la dirección  $\mathbf{s}_2^0$ , y así sucesivamente hasta acabar en el punto  $\mathbf{x}_n^0$  (la figura al final de esta sección ilustra el caso para dos dimensiones).

Paso 2.- Buscamos el punto particular  $\mathbf{x}_k^0$ , para el cual se ha obtenido una mejoría mayor de la función objetivo respecto al punto anterior  $\mathbf{x}_{k-1}^0$ . Definimos

dos magnitudes:

$$\Delta^k = [f(\mathbf{x}_{k-1}^0) - f(\mathbf{x}_k^0)].$$

$$\mu = \mathbf{x}_0^0 - \mathbf{x}_n^0.$$

Paso 3.- Determinamos:

$$f_t^0 = f(2\mathbf{x}_n^0 - \mathbf{x}_0^0). \quad (\text{A.3})$$

y llevamos a cabo dos comparaciones.

$$f_t^0 \geq f(\mathbf{x}_0^0),$$

$$(f(\mathbf{x}_0^0) - 2f(\mathbf{x}_n^0) + f_t^0) (f(\mathbf{x}_0^0) - f(\mathbf{x}_n^0) - \Delta) \geq \frac{\Delta (f(\mathbf{x}_0^0) - f_t^0)}{2}. \quad (\text{A.4})$$

Entonces la dirección no es una buena dirección de búsqueda y repetiríamos la búsqueda comenzando desde el punto  $\mathbf{x}_0^n$  como punto base. En caso contrario, se procede a incorporar la dirección al conjunto de direcciones de búsqueda, sustituyendo a la dirección que peor resultado hubiese obtenido.

En la nueva etapa de búsqueda conviene que la última dirección investigada (en la etapa de búsqueda unidireccional) sea  $\mu$ .

Las dos desigualdades anteriores comprueban, la primera, si se obtiene una mejora en la dirección al pasar del punto  $\mathbf{x}_0^0$  al punto  $\mathbf{x}_n^0$ , y la segunda, que la función descienda de manera pronunciada y no a través de una zona plana.

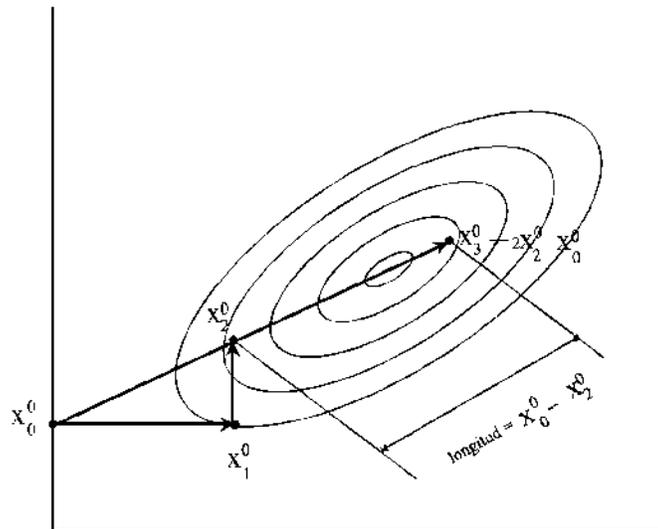


Figura A.2: Método de Powell.



# Apéndice B

## Buscador eficiente de vecinos

Durante la ejecución del proceso de relajación para los datos iniciales, el método del gradiente adaptado hereda uno de los más conocidos problemas presentes al trabajar con el método de SPH, y éste es el hecho de que al implementar el algoritmo para el caso  $N$  partículas, a medida que  $N$  crece el tiempo de ejecución del algoritmo hace ineficiente al mismo.

Esto concretamente se debe a que para un sistema discretizado de  $N$  partículas el método del gradiente requiere un total de  $N^2$  operaciones para calcular a iteración cada uno de los promedios a la SPH necesarios. Esto es; si por ejemplo, se trabajase en un sistema de 1000 partículas en 3 dimensiones, necesitaríamos realizar  $1000^2$  procesos para poder obtener el promedio de la densidad y de sus parciales en cada iteración.

Debido a esto, es necesario implementar un algoritmo capaz de encontrar a aquellos vecinos que nos permitan realizar los promedios a la SPH en menor tiempo (con un menor número de operaciones); esto es, necesitamos de un modo de calcular los promedios a la SPH sobre cada partícula con prácticamente solo las partículas vecinas que si aportan al cálculo de dicho promedio (aquellas que están dentro de su alcance  $h$ ).

A este proceso le llamaremos, *búsqueda eficiente de vecinos* y su importancia recaerá en que nos permitirá reducir el número de operaciones de  $N^2$  a solo  $N$ , reduciendo con ello considerablemente el tiempo de ejecución del proceso de relajación.

## B.1. Clasificación de las partículas

Para poder implementar el algoritmo de búsqueda eficiente de vecinos, lo primero que deberemos hacer será clasificar a las partículas. Esta clasificación consiste en asignar una *clave* a cada una de las  $N$  partículas con coordenadas  $r_a = (x_a, y_a, z_a)$  donde  $a = 1, \dots, N$ . Dicha clave nos permitirá hacer referencia específica a cada una de las partículas cuando así lo requiramos.

Para poder asignar una clave a las partículas, lo primero que haremos será delimitar la región en la que estas se encuentran y para ello construiremos la caja de dimensiones mínimas que contiene a todas las  $N$  partículas, será tal que sus aristas serán paralelas a los ejes coordenados.

Esto es, buscaremos los vértices de la caja tales que

$$\begin{aligned}x_{max} &= \max_{a=1, \dots, N} \{x_a, \} \\x_{min} &= \min_{a=1, \dots, N} \{x_a, \},\end{aligned}\tag{B.1}$$

y de igual modo para  $y_{min}, y_{max}, z_{min}, z_{max}$ .

Hecho esto subdividiremos la caja en un conjunto de pequeñas cajas que llamaremos celdas, las cuales contendrán en un futuro a cierto número de partículas a las que se les asignará una misma clave. Para calcular el número de celdas que contendrá dicha caja, habrá que calcular el número entero de subdivisiones de la caja por eje coordenado, esto es

$$\begin{aligned}C_x &= \frac{x_{max} - x_{min}}{h_{ens}}, \\C_y &= \frac{y_{max} - y_{min}}{h_{ens}}, \\C_z &= \frac{z_{max} - z_{min}}{h_{ens}},\end{aligned}\tag{B.2}$$

donde  $h_{ens}$  es el promedio de la distancia  $h$  evaluado con respecto a todas partículas. El número de celdas estará dado entonces por el producto  $C_x \cdot C_y \cdot C_z$ .

Una vez se ha subdividido la caja en cubos de dimensión  $h_{ens}$ , designaremos al conjunto de partículas contenidas en una misma celda como una *clase*, donde habría que recalcar que cada una de estas partículas pertenecientes a una misma clase tendrán una misma clave (la clave que designa a la clase o celda). A cada clase se le asignará una triada de números enteros  $(i_a, j_a, k_a)$  que nos indicarán la posición

de la celda a la que pertenece dicha clase; y eso se hará mediante

$$\begin{aligned} i_a &= 1 + \frac{x_a - x_{min}}{h_{ens}}, \\ j_a &= 1 + \frac{y_a - y_{min}}{h_{ens}}, \\ k_a &= 1 + \frac{z_a - z_{min}}{h_{ens}}, \end{aligned} \tag{B.3}$$

Cada uno de estos números enteros (índices de las celdas) podrán tomar valores enteros que van de 1 hasta al valor total de subdivisiones hechas a la caja con respecto al respectivo eje coordenado.

Finalmente, habrá que asignar una clave a cada una estas clases (o celdas), de tal manera que cada una de estas clases tenga una clave única por medio de la cual podamos referirnos a ella de forma precisa. Para esto, será importante construir una función que nos genere dichas claves y que aumente su valor de forma uniforme conforme los índices  $i_a$ ,  $j_a$  y  $k_a$  aumentan; lo cual se puede lograr mediante la ecuación:

$$clave(a) = i_a + C_x \cdot j_a + C_x \cdot C_y \cdot k_a. \tag{B.4}$$

Así, cada partícula tendrá una clave asignada a través de la celda a la que pertenezca. Por ejemplo, una partícula en la celda con coordenadas  $(i, j, k) = (1, 1, 1)$  tiene una clave dada por  $clave_{(1,1,1)} = 1 + C_x \cdot 1 + C_x \cdot C_y \cdot 1$ .

La función a través de la cual se generarán las claves es una función monótonamente creciente mientras las celdas se recorran de la siguiente forma:

- 0.- Iniciamos con  $i_a = 1, j_a = 1, k_a = 1$ .
- 1.- El índice  $i_a$  aumenta en 1 hasta que es igual a  $C_x$ .
- 2.- El índice  $j_a$  aumenta en 1.
- 3.- Se repiten los pasos 1 y 2 hasta que  $j_a$  es igual a  $C_y$ .
- 4.- El índice  $k_a$  aumenta en 1.
- 5.- Se repiten los pasos 1 a 4 hasta que  $k_a$  es igual a  $C_z$ .

## B.2. Ordenamiento de raíz.

Generalmente, el índice de las partículas puede no estar ordenado de tal forma que la función que genera las claves sea monótonamente creciente. Por lo cual, usaremos un procedimiento de reordenamiento llamado *ordenamiento de raíz*. Este, es un punto fundamental en el método de la búsqueda eficiente de vecinos ya que el

ordenamiento de raíz ha mostrado ser de orden  $N$ , heredando así esta propiedad a nuestro método del gradiente en el cálculo de los promedios a la SPH.

El ordenamiento de raíz aplicado a números enteros consiste en considerar la expresión binaria con lo cual se representa a los números que vamos a ordenar. Para ejemplificar su funcionamiento, consideremos la representación binaria de cinco números: 13, 22, 01, 54, 40.

001101	13
010110	22
000001	01
110110	54
101000	40

Como primer paso observemos el primer dígito de la derecha a la izquierda en las representaciones binarias de los números. Notamos así que el 13 presenta un 1, el 22 un 0, el 01 un 1, el 54 un 0 y el 40 un 0. Lo que se hace a continuación es tomar todos los elementos de la lista que presentan un 0 y los guardamos en el correcto orden de aparición, subsecuentemente se hace lo mismo para todos los que presentan un 1, con lo cual, pegando ambas listas comenzando por la de los números que presentan un 0, tendremos como resultado:

010110	22
110110	54
101000	40
001101	13
000001	01

A continuación haremos lo mismo con el lugar correspondiente al segundo dígito de la representación binaria de los números enlistados tomándolos de nuevo de derecha a izquierda. Ahora el 13 presenta un 0, el 22 un 1, el 01 un 0, el 54 un 1 y el 40 un 0. Después de ordenarlos de la misma forma descrita en el paso anterior tendremos como resultado:

101000	40
001101	13
000001	01
010110	22
110110	54

Una vez dominada la secuencia, continuaremos hasta terminar con los 6 dígitos con los que están compuestos los números de la lista a ordenar. Esto es:

Paso 3		Paso 4		Paso 5		Paso 6	
101000	40	000001	01	000001	01	000001	01
000001	01	010110	22	101000	40	001101	13
001101	13	110110	54	001101	13	010110	22
010110	22	101000	40	010110	22	101000	40
110110	54	001101	13	110110	54	110110	54

Tal como uno puede intuir, el resultado final del proceso es 01, 13, 22, 40, 54. Notemos que el proceso se ha llevado a cabo en 6 pasos, esto de acuerdo a la representación binaria, la cual es independiente del tamaño de la lista. Las operaciones internas al guardar en las listas de *ceros* y *unos* son las únicas que varían con el tamaño de la lista de elementos a ordenar. Hay que considerar también el hecho de que este ejemplo emplea enteros de *6-bits*; en caso de necesitar enteros de *32-bits* o más, entonces el número de operaciones aumentará a 32 o más.

### B.3. Censo de las partículas.

Hasta ahora se ha generado una lista de partículas y una lista de claves, pero aun no hemos referenciado explícitamente a cada partícula con respecto a la clave de la clase a la que pertenecen. Para hacer esto, comprobaremos la condición  $clave_{actual} = clave(a)$ , con  $a$  que va de 1 hasta  $n$  y donde  $clave(a)$  representa a la lista generada y  $clave_{actual}$  está dada por?

$$clave_{actual} = i_a + C_x \cdot j_a + C_x \cdot C_y \cdot k_a. \quad (B.5)$$

Si la condición antes mencionada se cumple, entonces la partícula  $a$ -ésima pertenece a la clase designada con  $clave(a)$  y sucesivamente pertenecerán a la misma clase todas aquellas partículas para las que la condición se cumpla. Una vez que la condición sea falsa, habremos encontrado a una partícula perteneciente a otra clase (celda). Durante el proceso, cada vez que se viole la condición, será importante guardar los siguientes datos:

- 1.- Índice de la  $a$ -ésima partícula con la que inicia una clase e índice de la  $a$ -ésima partícula con la que termina.
- 2.- Indicador lógico para identificar celdas ocupadas y celdas vacías.

Al finalizar el ciclo sobre las  $N$  partículas, tendremos que el número de veces que no se cumplió la condición, nos dará el número total de clases que tiene la configuración, entonces el índice de las clases va de  $i = 1, \dots, Nclases$ .

## B.4. Búsqueda de las partículas.

En las secciones anteriores se ha realizado una descripción de los pasos a seguir a fin de agilizar el proceso consistente en realizar los promedios necesarios para el método de relajación mediante una búsqueda eficiente de los vecinos de cada partícula.

Ahora, dada la  $a$ -ésima partícula del total  $N$ , podemos extraer la clave que está asociada a ella,  $clave(a) = q$ , y a partir de la misma, podemos encontrar la celda en la que se ubica esta partícula haciendo uso de la expresión:

$$\begin{aligned} i &= \text{mod}(q, C_x), \\ j &= \text{mod}\left(\frac{q-i}{C_x}, C_y\right), \\ k &= \frac{q-i-C_x \cdot j}{C_x \cdot C_y}. \end{aligned} \quad (\text{B.6})$$

En los casos en que  $i$ ,  $j$  o  $k$  arrojen un valor igual a 0, se les deberá asignar respectivamente un valor dado por  $C_x, C_y, C_z$ .

Una vez que tenemos las coordenadas  $(i, j, k)$  de la celda en la que se ubica la  $a$ -ésima partícula, obtendremos las coordenadas  $(i_{vecino}, j_{vecino}, k_{vecino})$  de sus celdas vecinas mediante un ciclo anidado:

$$N_{celdas}(a) = \{(i_{vecino}, j_{vecino}, k_{vecino}) \mid i_{vecino} = i + r, j_{vecino} = j + s, k_{vecino} = k + t\}, \quad (\text{B.7})$$

con  $i_{vecino}, j_{vecino}, k_{vecino} = -Rango, Rango$   
donde

$$Rango = 2\left(\frac{H_i}{h_{ens}} + 1\right) \quad (\text{B.8})$$

Para referenciar el índice de la clase, usaremos la expresión

$$Index = i_{vecino} + C_x \cdot (j_{vecino} - 1) + C_x \cdot C_y \cdot (k_{vecino} - 1). \quad (\text{B.9})$$

Cada una de estas celdas (clases) tiene asociada una lista ordenada que se construyó en el censo. A través de ésta, podemos saber si existen partículas en la celda y de existir podemos referenciar directamente a cada una de ellas.

Con ello, habremos encontrado un subconjunto de partículas que serán las partículas vecinas  $j$ -ésimas a través de las cuales calcularemos los promedios necesarios con respecto a la partícula  $a$ -ésima.

Debemos considerar que ciertos casos donde la partícula  $a$ -ésima esté en las fronteras; para estos casos, no podremos explorar todas las celdas vecinas ya que la mayoría de estas no existirán, por ello deberemos confirmar que para las coordenadas  $(i, j, k)$  de cada una de las  $N_{celdas}$  se cumpla que

$$\begin{aligned} i &\leq C_x, j \leq C_y, k \leq C_z, \\ i &\geq 1, j \geq 1, k \geq 1. \end{aligned} \tag{B.10}$$

## B.5. Tiempo de procesamiento.

El fundamento para incorporar a nuestro código el uso de un buscador eficiente de vecinos no es otro más que reducir considerablemente el tiempo de cómputo que él mismo necesita para llevar a acabo la tarea que se le solicita.

En la figura B.1 se comparan los tiempos de procesamiento entre el código sin buscador de vecinos y la versión con el mismo incorporado. Para el análisis se han llevado a cabo mediciones comparativas entre ambas versiones bajo diferentes configuraciones de partículas empezando por configuraciones sencillas para luego agregarles cada vez mas partículas.

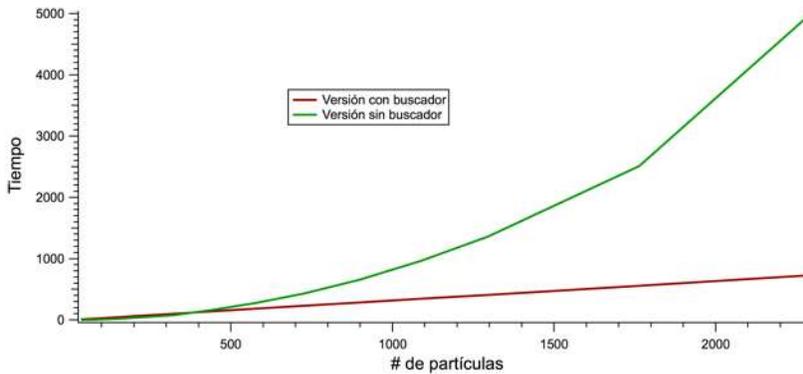


Figura B.1: Comparativa del tiempo de procesamiento entre la versión del código con buscador de vecinos y la versión sin buscador.

La gráfica nos muestra que para el caso con el buscador de vecinos implementado tendremos un comportamiento lineal en el tiempo de procesamiento a medida que aumentemos el número de partículas mientras que para la versión sin buscador el tiempo de procesamiento tendrá un comportamiento cuadrático en su crecimiento.

Con esto queda corroborado lo importante y eficiente de implementar el buscador de vecinos para el proceso de relajación del sistema.

# Apéndice C

## Anticolisionador.

Dado el problema de los pares de partículas que se mantienen unidos comportándose como una sola partícula, el cual se describió en la sección ‘Análisis preliminar del método’ se puede tomar la idea inicial de dar una patada al sistema tras caer en dicho problema para crear una subrutina que funja como una especie de barrera (anticolisionador) que haga que las partículas se repelan las unas a las otras cuando estén lo suficientemente cerca bajo un criterio asignado.

Para construir este sistema de anticolisionamiento de forma sencilla y eficaz consideraremos el aplicar aquella patada a las partículas pegadas pero no con el fin de separarlas bruscamente a fin de romper su configuración actual sino con el fin de evitar que se peguen demasiado la una a la otra. Por tanto el anticolisionador no será una herramienta que se emplee una vez y se deseche, sino que será una herramienta a ejecutar a lo largo de todo el proceso de relajación.

La barrera virtual que separe a las partículas, estará a una distancia de la partícula  $i$ -ésima proporcional a la separación ideal que debiera haber entre ella y las partículas  $j$ -ésimas vecinas en una configuración ya relajada. Para los casos en se usó una densidad constante bastará con simplemente evaluar cual es esa distancia ideal y elegir un valor idóneo para dicha barrera; que en la practica fue de  $1/8$  de la separación idónea entre las partículas.

Para el caso de la configuración con densidad radial, la distancia ideal para la barrera se construirá a partir de la longitud  $h$ . Del buscador de vecinos sabemos que las celdas que empleamos para indexar a las partículas tienen una longitud  $h$  promedio cada una, por tanto podemos tomar este valor  $h$  promedio para construir la distancia para la barrera bajo el criterio que consideremos mas adecuado.

Hasta ahora hemos especificado solo la barrera que es la distancia con respecto a la cual sabremos si alguna partícula  $j$ -ésima se ha acercado demasiado a una cierta

partícula  $i$ -ésima, pero no hemos especificado como haremos dicha evolución durante el proceso. Es aquí donde nuestro buscador eficiente de vecinos antes construido será de nuevo fundamental.

A partir del buscador podemos modificar su estructura para hacer una versión mas ligera que solo busque para una partícula  $i$ -ésima las partículas vecinas  $j$ -ésimas tanto en la celda donde se encuentra como en las celdas vecinas inmediatas. Lo anterior considerando el hecho de que pueda darse el caso en que la partícula  $i$ -ésima a analizar se encuentra en la frontera de su celda y muy próxima a otra partículas ubicada en una celda vecina.

Con el buscador de vecinos se podrá mantener una constante búsqueda de partículas que se junten demasiado bajo un criterio dado por la barrera antes descrita. Lo que resta ahora es saber que hacer una vez que se encuentren dos partículas que se han acercado demasiado. La noción de patada a las partículas tiene aquí un papel importante pero con un enfoque diferente al antes considerado.

Dado que lo que deseamos es que las partículas busquen su posición idónea durante el proceso de relajación pero sin que vayan conformando pares de partículas, no necesitamos aventar muy lejos a las partículas que se están juntando la con respecto de la otra. Bastará con simplemente darles un pequeño empujoncito para separarlas en direcciones contrarias. Si consideramos que las partículas volverán a acercarse conforme el proceso siga avanzando y si consideramos que a su vez el anticolidionador las irá separando, podemos establecer que con la implementación del anticolidionador el proceso de relajación de las partículas se llevará a cabo de la misma forma en que se había llevado a cabo antes pero con la diferencia de que ahora las partículas que intenten pegarse las unas a las otras para conformar pares se verán obligadas a mantenerse constantemente separadas por una barrera.

Finalmente, será importante analizar el tiempo de procesamiento que implica la incorporación de esta herramienta a nuestro proceso de relajación y para ello debemos realizar una comparativa entre el tiempo invertido por el proceso de relajación con el anticolidionador y sin él. La figura C.1 presentada a continuación ilustra justamente dicha comparación y deja en claro que prácticamente no hay diferencia en el tiempo de procesamiento en el hecho de usar o no el anticolidionador.

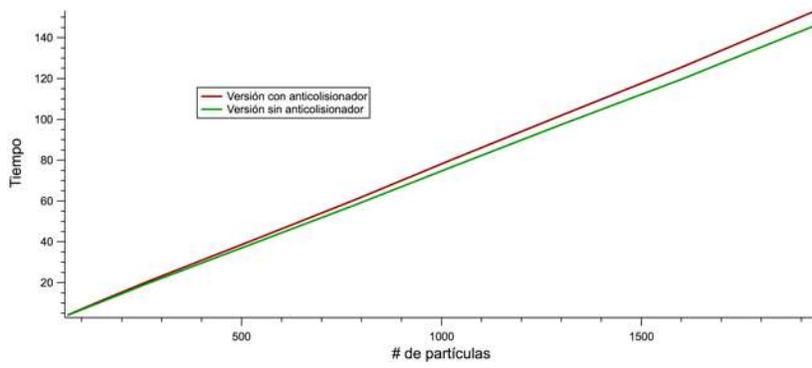


Figura C.1: Comparativa del tiempo de procesamiento entre la versión del código con Anticollisionador y la versión sin él.



# Bibliografía

- [1] Ayal, S., Piran, T., Oechslin, R., Davies, M. B., & Rosswog, S. (2001). Post-Newtonian smoothed particle hydrodynamics. [Article]. *Astrophysical Journal*, 550(2), 846-859.
- [2] Liu, G. R., & Liu, M. B. (2003). *Smoothed particle hydrodynamics : a meshfree particle method*. Singapore ; London: World Scientific.
- [3] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.*, 30, 543-574.
- [4] Atzberger, P. J. *The Monte-Carlo Method*.
- [5] Sawilowsky, S. S. (2003). You Think Youve Got Trivials?. *Journal of Modern Applied Statistical Methods*, 2(1), 218-225.
- [6] Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, 110,399406.
- [7] Rosswog, S. (2009). Astrophysical smooth particle hydrodynamics. [Review]. *New Astronomy Reviews*, 53(4-6), 78-104.
- [8] Monaghan, J. J. (2005). Smoothed particle hydrodynamics. [Review]. *Reports on Progress in Physics*, 68(8), 1703-1759.
- [9] Franklin, J. (2013). *Computational methods for physics*. Cambridge: Cambridge University Press.
- [10] Hernández, G. (2006). *Métodos Clásicos de Optimización para Problemas No-Lineales sin Restricciones*. U. de Chile.
- [11] Sánchez, D. (2000). *Estudio comparativo de diversos métodos de optimización de geometrías moleculares en coordenadas cartesianas*. Universidad de Cádiz.

- [12] Hernández, R. (2012). Sobre una nueva búsqueda lineal tipo armijo y su relación con el método de region de confianza. Universidad centroccidental "Lisandro Alvarado".
- [13] Fletcher, R., & Reeves, M. (1964). Function minimization by conjugate gradient. *Computer Journal*, 7(2), 149-154.
- [14] Rosswog, S. (2010). Conservative, special-relativistic smoothed particle hydrodynamics. [Article]. *Journal of Computational Physics*, 229(22), 8591-8612.
- [15] Birgin, E. G., Martnez J. M. & Raydan, M. (2013). Spectral Projected Gradient methods: Review and Perspectives. *Journal of Statistical Software*, 50(2).
- [16] Martinez, J. M., Pilotta, E. A., & Raydan, M. (2005). Spectral gradient methods for linearly constrained optimization. [Article]. *Journal of Optimization Theory and Applications*, 125(3), 629-651.
- [17] Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(2), 155-162.