

Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Aplicaciones del método smoothed particle
hydrodynamics (SPH) en la dinámica de gases

Autor: José Ignacio Pérez Puig

Tutor: Miguel Pérez-Saborid Sánchez-Pastor

Dep. Ingeniería Aeroespacial y Mecánica de Fluidos
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017



Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Aplicaciones del método smoothed particle hydrodynamics (SPH) en la dinámica de gases

Autor:

José Ignacio Pérez Puig

Tutor:

Miguel Pérez-Saborid Sánchez-Pastor

Profesor titular

Dep. Ingeniería Aeroespacial y Mecánica de Fluidos

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Trabajo Fin de Grado: Aplicaciones del método smoothed particle hydrodynamics (SPH) en la dinámica de gases

Autor: José Ignacio Pérez Puig

Tutor: Miguel Pérez-Saborid Sánchez-Pastor

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

A mi familia, amigos y profesores.

Agradecimientos

Deseo expresar mi agradecimiento a todas las personas que me han acompañado durante estos años, en especial a mis padres por su apoyo incondicional y a mis amigos, los cuales han conseguido hacer el camino más fácil disfrutando de cada instante.

También me gustaría agradecer a los profesores que me han formado durante estos años, especialmente a Don Miguel Pérez-Saborid Sánchez-Pastor por el gran apoyo recibido durante los últimos meses.

José Ignacio Pérez Puig

Sevilla, 2017

Resumen

El objetivo de este proyecto es la implementación del método Smoothed Particle Hydrodynamics (SPH) en MATLAB, para el estudio de la dinámica de gases. Se presentarán los fundamentos teóricos y ecuaciones del método que permiten la resolución numérica de diversos problemas en el ámbito de la ingeniería y de la ciencia

El método SPH se diferencia del resto por ser libre de malla, además del uso de partículas para la resolución numérica de problemas fluido-dinámicos. La principal ventaja es que ofrece la posibilidad de entender la física del problema al mismo tiempo que se obtienen los resultados para problemas reales, los cuales no son abordables analíticamente.

En el proyecto se aplica el método SPH en el estudio del problema unidimensional de ondas de choque en un tubo donde existe una discontinuidad en las propiedades físicas. Además del estudio de la evolución con respecto al tiempo de un modelo simplificado de estrella, viendo como a partir de una configuración inicial aleatoria alcanza el estado de equilibrio.

Abstract

The aim of this project is to program the Smoothed Particle Hydrodynamics method (SPH) in MATLAB in order to study the fluid dynamics problems. It presents an overview on the SPH theory and formulation, that allows us to reach the numerical solution of several problems in engineering and science.

SPH is a meshfree and particle-based method for computational fluid dynamics. The main advantage of this method is the chance of understanding the physical concepts of the problem and at the same time, obtain solutions to real problems without analytical solution.

In this project, SPH will be applied to one-dimensional shock tube problem where shocks are due to a discontinuity of physics fields. Also, to study the time evolution of a toy star model and find its equilibrium state.

Índice

Agradecimientos	9
Resumen	11
Abstract	13
Índice	14
Índice de Figuras	¡Error! Marcador no definido.
1 Introducción	16
1.1 <i>Historia SPH</i>	17
1.2 <i>Ventajas del SPH frente a otros métodos</i>	18
1.3 <i>Objetivos y Estructura del trabajo</i>	19
2 Fundamentos teóricos	20
2.1 <i>Aproximación SPH</i>	20
2.1.1 <i>Aproximación mediante función kernel</i>	20
2.1.2 <i>Aproximación mediante partículas</i>	24
2.2 <i>Función kernel</i>	26
2.3 <i>Técnicas para obtener la formulación SPH</i>	28
2.4 <i>Interpolación a partir de las partículas</i>	29
3 SPH aplicado a dinámica de fluidos	30
3.1 <i>Ecuaciones de Euler en SPH</i>	30
3.1.1 <i>Ecuación continuidad</i>	31
3.1.2 <i>Ecuación de la aceleración</i>	31
3.1.3 <i>Viscosidad artificial</i>	32
3.1.4 <i>Ecuación de la energía</i>	33
3.2 <i>Integración en el tiempo</i>	33
4 Tubo de choque	34
4.1 <i>Descripción del problema</i>	34
4.2 <i>Resultados</i>	35
4.3 <i>Códigos Matlab</i>	38
5 Modelo simplificado de estrella	43
5.1 <i>Descripción del problema</i>	43
5.2 <i>Resultados</i>	44
5.3 <i>Códigos Matlab</i>	55
6 Conclusiones y desarrollos futuros	62
Referencias	63

1 INTRODUCCIÓN

“Lo que sabemos es una gota de agua; lo que ignoramos es el océano”

-IsaacNewton -

Las simulaciones mediante ordenadores se han convertido, durante estos años, en una herramienta muy importante en la resolución de problemas complicados dentro del ámbito de la ingeniería y la ciencia. La gran mayoría de los problemas fluido-mecánicos no tienen solución analítica y es necesario la utilización de métodos numéricos para su estudio.

Estas simulaciones desempeñan un papel importante, sirviendo de prueba para numerosas teorías, que de otra manera serían muy difíciles de demostrar. Métodos numéricos basados en mallados tales como el método de las diferencias finitas, el método de los elementos finitos... han sido aplicados ampliamente en diferentes áreas de la dinámica de fluidos computacional a la vez que en la mecánica de sólidos.

Una de las características más notables de los métodos basados en mallas, es la división de un medio continuo en pequeños subdominios discretos, mediante un proceso de discretización o mallado. De esta manera, las ecuaciones que gobiernan el problema pueden ser convertidas en un conjunto de ecuaciones algebraicas, donde las incógnitas corresponden a los valores de las variables en los nodos.

A pesar de la gran utilidad, estos métodos numéricos presentan numerosas dificultades en otros aspectos, limitando así su aplicación en ciertos tipos de problemas. Las mayores dificultades se presentan en el uso del mallado, el cual puede llevar a lidiar con problemas en el uso de superficies libres, contorno deformable, interfaz móvil, grandes deformaciones...

Por otro lado, en problemas con geometrías complicadas, la generación de una malla de calidad se ha convertido en una tarea complicada, costosa que requiere bastante tiempo.

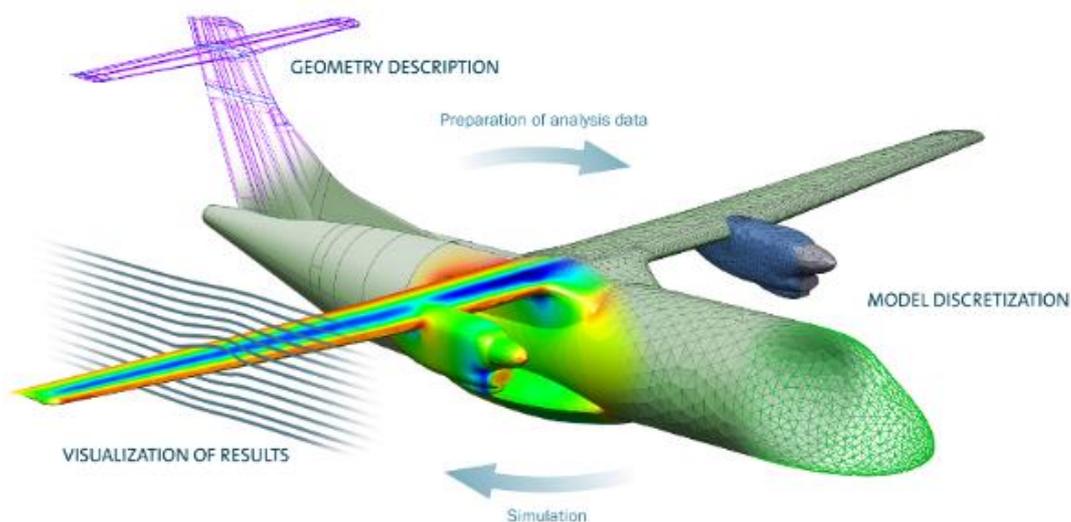


Figura (1-1)

Estas dificultades que presentan los métodos son especialmente evidentes en la simulación de fenómenos hidrodinámicos tales como explosiones e impactos a grandes velocidades. Durante el proceso de la explosión existen grandes deformaciones, interfaces móviles, superficies libres... En los impactos a grandes velocidades, además, tiene lugar la propagación de ondas de choque.

Por último, los métodos numéricos basados en mallas no son recomendables para situaciones donde el objeto de estudio es un conjunto discreto de partículas físicas en lugar de un medio continuo. Un ejemplo típico es el correspondiente a la interacción de estrellas en la astrofísica, movimiento de millones de átomos en estado de equilibrio o inequilibrio.

Durante los últimos años, se ha investigado en una nueva generación de métodos computacionales, métodos libres de malla. La idea de estos métodos es proporcionar soluciones precisas y estables a las ecuaciones diferenciales que describen el movimiento, con cualquier tipo de condiciones de contorno.

Uno de estos métodos es el Smoothed Particle Hydrodynamics (SPH), objeto de estudio en este proyecto.

1.1 Historia SPH

El método SPH originalmente fue inventado para la resolución de problemas astrofísicos de tres dimensiones (Lucy 1977; Gingold and Monaghan, 1977) en el espacio abierto. Dado que el movimiento colectivo de dichas partículas es similar al del movimiento de un líquido o un gas, se puede modelar mediante las ecuaciones clásicas de Newton para la hidrodinámica.



Figura (1-2)

Aunque los métodos tradicionales basados en mallas, como el método de las diferencias finitas o el método de los elementos finitos ya existían, y en general se encontraban más desarrollados que el SPH, presentaban dificultades a la hora de manejar complejos fenómenos como se ha visto antes.

Esta circunstancia motivó a los investigadores a buscar alternativas para la resolución de este tipo de problemas, convirtiéndose el método SPH en una buena opción.

A día de hoy, SPH es usado en una gran cantidad de áreas, desde simulaciones de estrellas binarias, supernovas, hasta el estudio de la evolución del universo.

1.2 Ventajas del SPH frente a otros métodos

En el método SPH el sistema de estudio es representado por un conjunto de partículas, que tienen propiedades de un material e interactúan entre ellas dentro de un rango controlado por una función de peso o función kernel. La discretización de las ecuaciones que gobiernan el movimiento se basa en este sistema de partículas discretas.

Se emplean numerosas formulaciones basadas en partículas para el cálculo de la densidad, velocidad y aceleración del fluido. La presión se obtiene mediante la ecuación de estado, de esta manera la aceleración se obtiene del gradiente de presiones y la densidad. Para el caso de fluidos viscosos, los efectos viscosos sobre la aceleración de la partícula también son incluidos.

Como método Lagrangiano basado en partículas, SPH conserva exactamente la masa. El movimiento del fluido es representado por el movimiento de dichas partículas.

El SPH presenta algunas ventajas especiales sobre los métodos numéricos tradicionales basados en mallas, anteriormente presentados. Estas ventajas se pueden resumir en los siguientes puntos:

- Se trata de un método ideal para el modelado de superficies libres y problemas fluido dinámicos con interfaces. Las superficies libres, contornos móviles e interfaces de material pueden ser monitorizadas durante la simulación sin importar el complejo movimiento de las partículas.

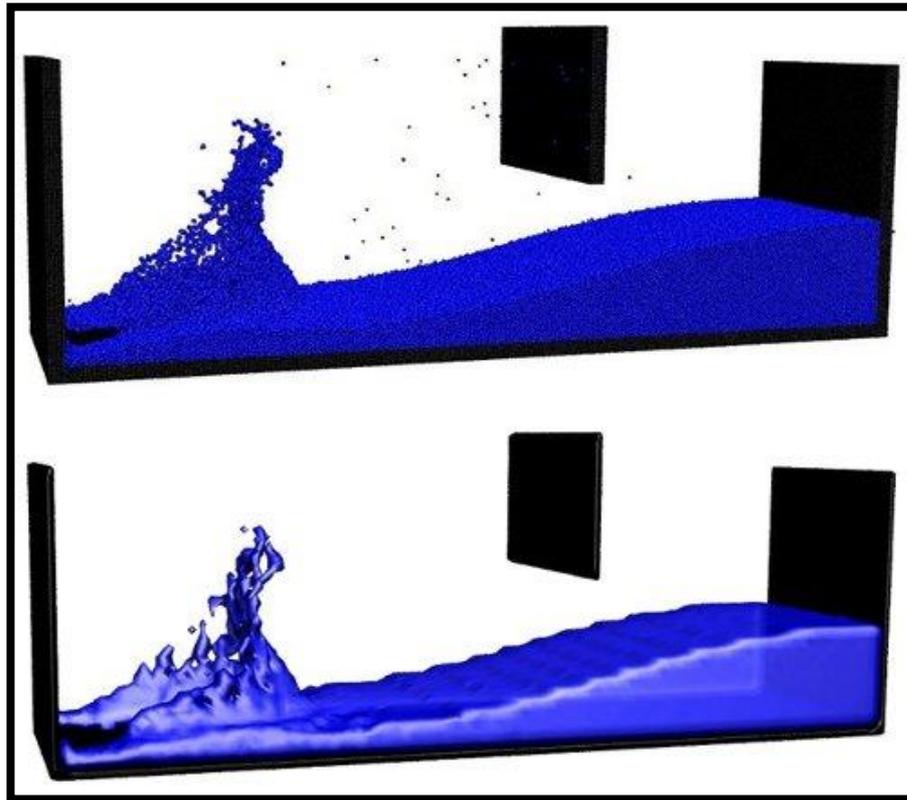


Figura (1-3)

- SPH es un método como ya se ha comentado que no emplea un mallado. Este hecho permite un manejo directo y sencillo de las grandes deformaciones, ya que la conectividad entre las partículas es generada como parte de simulación y puede variar a lo largo del tiempo.
- En el método, la partícula representa un volumen finito en la escala continua. Similar al método de dinámica molecular que emplea el uso de partículas para representar un átomo o molécula en nano-escalas. Por tanto, es natural extender el uso del SPH a escalas menores, especialmente en ámbitos como la biofísica o biomedicina.

- SPH es apropiado para para problemas donde el objeto bajo consideración no es continuo. Por ejemplo, en bio-ingeniería, nano-ingeniería o astrofísica.

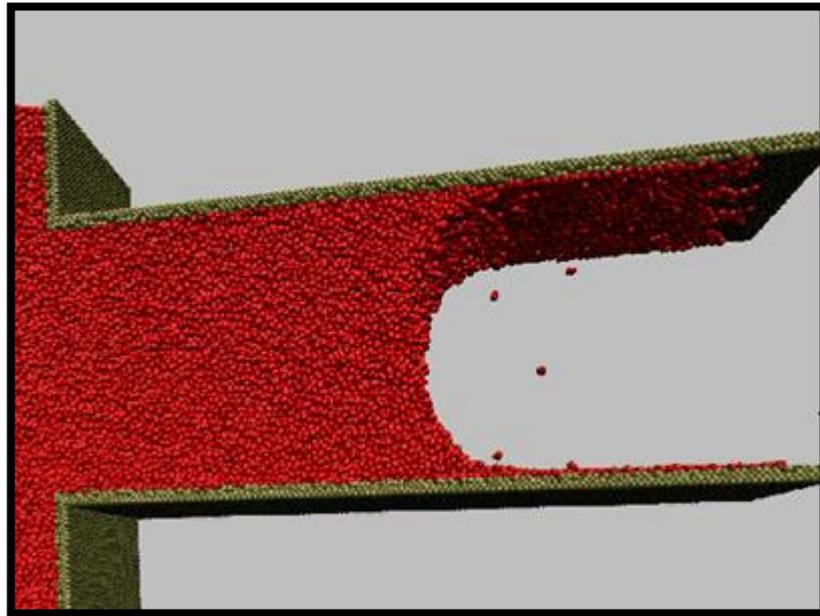


Figura (1-4)

- La implementación numérica es mucho más simple y el estudio de sistemas tridimensionales es mucho más natural que en el caso de los métodos tradicionales.

1.3 Objetivos y Estructura del trabajo

El Objetivo de este Trabajo de fin de Grado es doble: presentar las bases teóricas del método SPH e implementar el método en MATLAB para resolución de diversas aplicaciones de dinámica de gases. Se trata del primer trabajo realizado de este método en la Escuela Superior de Ingeniería de Sevilla y pretende ser un punto de partida para futuros trabajos e aplicaciones del SPH.

El método se aplicará en el estudio del problema unidimensional de ondas de choque en un tubo donde existe una discontinuidad en las propiedades físicas (Shock Tube). Además del estudio de la evolución con respecto al tiempo de un modelo simplificado de estrella y como alcanza el estado de equilibrio.

Con estos objetivos por cumplir, el trabajo se estructurará en seis capítulos más, además del presente.

En el siguiente capítulo se van a exponer los fundamentos teóricos sobre los que se basa el método SPH a implementar. Seguidamente, se explicará cómo se formulan las ecuaciones de Euler dentro del método. En los capítulos 4 y 5, se analizarán los problemas del tubo de choque y modelo simplificado de estrella, presentando los resultados obtenidos en las simulaciones junto a los códigos empleados. Finalmente, las conclusiones y futuros desarrollos serán expuestos en el capítulo 6.

2 FUNDAMENTOS TEÓRICOS

*“Las matemáticas son el lenguaje en el que
Dios ha escrito el Universo”
-Galileo Galilei-*

En el capítulo 1, se ha visto que el método Smoothed Particle Hydrodynamics (SPH) presenta características diferentes a los métodos tradicionales empleados en la resolución numérica de problemas dinámicos de fluidos. En este capítulo se presentarán los conceptos y fundamentos teóricos esenciales para la formulación del método.

Se analizará la estrategia del método SPH en la aproximación de variables mediante la representación integral de una función y la discretización en partículas.

También se expondrán técnicas usadas en la derivación de las formulaciones SPH e interpolación en puntos diferentes a las partículas.

2.1 Aproximación SPH

La gran mayoría de problemas hidrodinámicos se encuentran en forma de ecuación en derivadas parciales (EDP) de variables tales como densidad, velocidad, energía... La obtención de una solución analítica para este conjunto de ecuaciones raramente es posible, exceptuando casos simplificados.

Para la resolución, primero se discretiza el dominio donde las EDPs están definidas. Una vez realizado, es necesario un método que facilite una aproximación de las funciones y sus derivadas en cualquier punto del dominio. Esta aproximación es aplicada al conjunto de EDPs llegando a ecuaciones diferenciales ordinarias (EDO) en forma discreta. Por último, se resuelven usando algún método de integración basado en diferencias finitas.

En el caso del método SPH, el dominio se representa mediante un conjunto de partículas arbitrariamente distribuidas. Para la obtención de la formulación SPH existen básicamente dos pasos. El primer paso es representar una función y sus derivadas como una integral, este paso se suele denominar aproximación mediante la función kernel. El segundo paso consiste en la aproximación mediante partículas. En este paso, el conjunto de EDOs es resuelto mediante la discretización del dominio con una distribución inicial de partículas donde se aproximan las variables.

2.1.1 Aproximación mediante función kernel

La representación integral de una función consiste en la aproximación mediante la suma de los valores correspondientes a las partículas cercanas, reduciendo la aproximación de la función a un punto o partícula.

El concepto de la representación integral de una función $f(\mathbf{x})$ comienza con la siguiente identidad

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}', \quad (1)$$

donde f es función del vector de posición tridimensional \mathbf{x} . Por otro lado, $\delta(\mathbf{x} - \mathbf{x}')$ es la función delta de Dirac dada por la siguiente igualdad

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} 1 & \mathbf{x} = \mathbf{x}' \\ 0 & \mathbf{x} \neq \mathbf{x}' \end{cases} \quad (2)$$

La ecuación (1) implica que una función puede ser representada en forma integral. Debido al uso de la función delta de Dirac, la integral representada en la ecuación (1) es exacta, siempre y cuando $f(\mathbf{x})$ este definida y sea continua en el volumen Ω .

Si reemplazamos la función delta de Dirac por una función $W(\mathbf{x} - \mathbf{x}', h)$, la representación integral de $f(\mathbf{x})$ viene dada por

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}', \quad (3)$$

donde W es la llamada función kernel. Dentro de la función, h es el parámetro llamado smoothing length, el cual define el área de influencia de la función kernel. Una vez se ha sustituido la función delta de Dirac, la representación integral de la ecuación (3) solo puede ser una aproximación. Por tanto, la ecuación se reescribe como

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}'. \quad (4)$$

La función W normalmente es una función par, por razones que se darán posteriormente. Además, debe satisfacer un cierto número de condiciones, la primera de ellas es la condición de normalización

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' = 1. \quad (5)$$

Esta condición es también llamada la condición unidad, dado que la integración de la función kernel en el dominio ha de ser igual a la unidad.

La segunda condición es la propiedad de delta de Dirac cuando el smoothing length se aproxima a cero

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

La tercera condición es la condición de compacidad

$$W(\mathbf{x} - \mathbf{x}', h) = 0 \quad \text{si } |\mathbf{x} - \mathbf{x}'| > kh, \quad (7)$$

siendo k es una constante relacionada con la función kernel para el punto \mathbf{x} y que define el área efectiva de la función. Esta área efectiva es llamada el dominio de soporte de la función kernel en el punto \mathbf{x} . Usando esta condición de compacidad, la integración sobre el dominio completo se reduce a la integración sobre el dominio de soporte de la función kernel.

Esta aproximación tiene una precisión de orden h^2 o de segundo orden. Los errores en la representación integral se pueden estimar mediante el uso de una serie de Taylor de $f(\mathbf{x}')$ en torno a \mathbf{x} . Si $f(\mathbf{x})$ es derivable y continua, tenemos

$$\begin{aligned} \langle f(\mathbf{x}) \rangle &\geq \int_{\Omega} [f(\mathbf{x}) + f'(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + r((\mathbf{x}' - \mathbf{x})^2)] \times W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \\ &= f(\mathbf{x}) \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \\ &\quad + f'(\mathbf{x}) \int_{\Omega} (\mathbf{x}' - \mathbf{x}) \times W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + r(h^2). \end{aligned} \quad (8)$$

Si W es una función par con respecto a \mathbf{x} y $(\mathbf{x}' - \mathbf{x})$, entonces $W(\mathbf{x} - \mathbf{x}', h)$ debe ser una función impar. Por tanto, se obtiene que

$$\int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0. \quad (9)$$

Usando la condición de unidad, y las anteriores ecuaciones llegamos a la siguiente expresión

$$\langle f(\mathbf{x}) \rangle = f(\mathbf{x}) + r(h^2). \quad (10)$$

Quedando claro que la aproximación mediante el kernel en el método SPH de una función arbitraria tiene una precisión de segundo orden.

La aproximación de una derivada espacial $\nabla \cdot f(\mathbf{x})$ se obtiene simplemente sustituyendo $f(\mathbf{x})$ por $\nabla \cdot f(\mathbf{x})$ en la ecuación (4), llegando a

$$\langle \nabla \cdot f(\mathbf{x}) \rangle = \int_{\Omega} [\nabla \cdot f(\mathbf{x}')] W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (11)$$

La divergencia en la integral es con respecto a la coordenada prima. Considerando la siguiente igualdad

$$[\nabla \cdot f(\mathbf{x}')] W(\mathbf{x} - \mathbf{x}', h) = \nabla \cdot [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)] - f(\mathbf{x}') \cdot \nabla [W(\mathbf{x} - \mathbf{x}', h)], \quad (12)$$

se obtiene la siguiente ecuación

$$\nabla \cdot f(\mathbf{x}) = \int_{\Omega} \nabla \cdot [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)] d\mathbf{x}' - \int_{\Omega} f(\mathbf{x}') \cdot \nabla [W(\mathbf{x} - \mathbf{x}', h)] d\mathbf{x}'. \quad (13)$$

La primera integral puede ser transformada mediante el teorema de la divergencia en una integral sobre la superficie S del dominio de integración Ω

$$\langle \nabla \cdot f(\mathbf{x}) \rangle = \int_S f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \cdot \vec{n} d\mathbf{x}' - \int_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (14)$$

donde \vec{n} es el vector normal de la superficie S . Desde que la función kernel es normalmente definida para tener soporte compacto, el valor de W en la superficie de la anterior integral es cero en SPH. Por consiguiente, la aproximación mediante el kernel de las derivadas se puede escribir como

$$\langle \nabla \cdot f(\mathbf{x}) \rangle = - \int_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (15)$$

La aproximación de la derivada en el SPH permite que el gradiente sea determinado a partir de los valores de la función y de las derivadas de la función kernel, en lugar de los valores de la derivada de la propia función.

La aproximación de derivadas de mayor orden se obtiene de una manera similar, sustituyendo $f(\mathbf{x})$ con la correspondiente derivada en la ecuación (4), usando integración por partes, el teorema de la divergencia y transformaciones triviales. Siguiendo análisis similares basados en series de Taylor, es fácil demostrar que la aproximación de las derivadas es también de segundo orden de precisión.

Las aproximaciones mediante kernel en el método SPH son de segundo como ver en la siguiente figura (2-1).

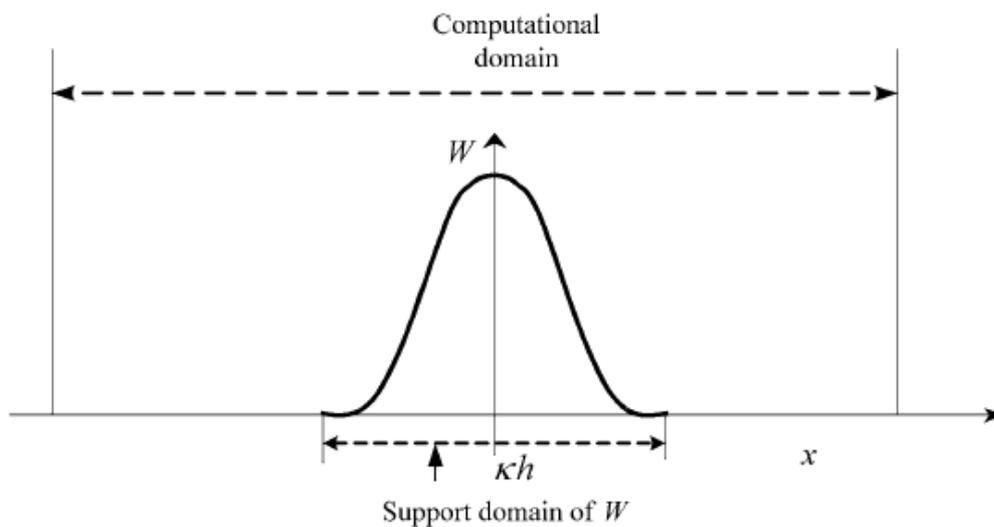


Figura (2-1)

Sin embargo, para algunos escenarios no se cumple este hecho. Por ejemplo, problemas donde el dominio de soporte intersecta el dominio del problema. En estos casos la función W es truncada por el contorno y la integración de la función kernel en el dominio ya no es igual a la unidad.

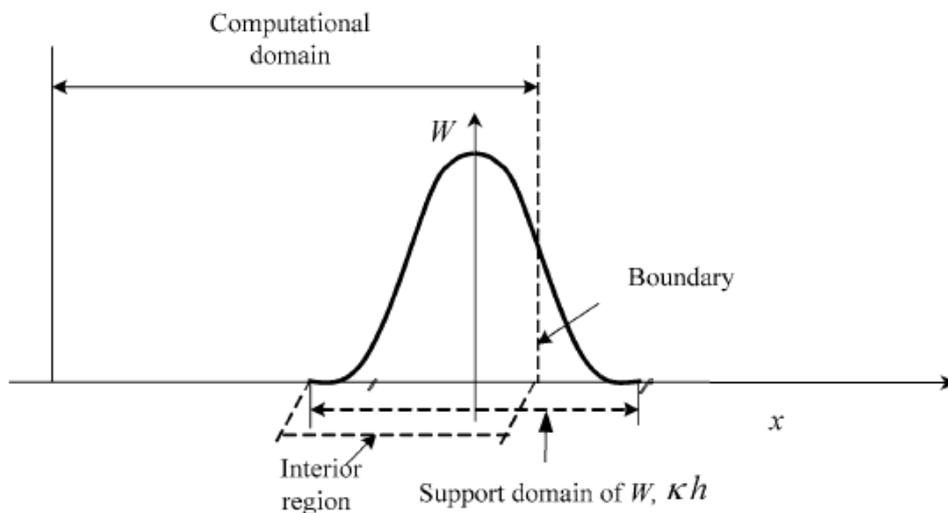


Figura (2-2)

2.1.2 Aproximación mediante partículas

En el método SPH, el sistema al completo es representado mediante un número finito de partículas, donde cada una de ellas tiene masa propia y ocupa una zona del espacio.

La representación integral mediante la función kernel se puede convertir a una forma discreta siendo la suma del valor de las partículas en el dominio. Esta discretización es conocida como la aproximación mediante partículas en el método SPH.

Considerando un dominio Ω formado por un conjunto de partículas, arbitrariamente distribuidas. El estado del sistema en cada instante es representado por estas partículas, cada una con sus propiedades. Estas partículas no son únicamente para la integración o interpolación, sino también para representar el material.

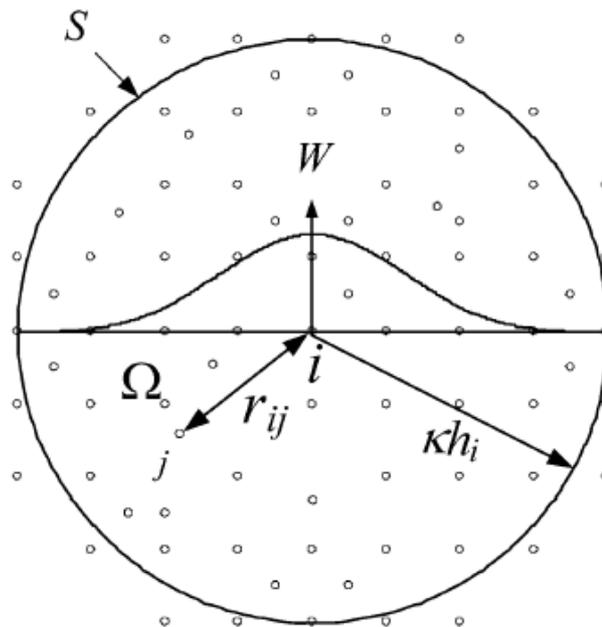


Figura (2-3)

La función W es usada para la aproximación de las variables de la partícula i usando la suma ponderada sobre las partículas j , dentro del dominio de soporte con una reducción de distancia κh_i .

El volumen infinitesimal $d\mathbf{x}'$ en las anteriores integrales de una partícula j , se reemplaza por el volumen infinitesimal de la partícula ΔV_j , a través de la siguiente expresión:

$$m_j = \Delta V_j \rho_j, \quad (16)$$

donde ρ_j es la densidad de la partícula j ($= 1, 2, \dots, N$) y N es el número de partículas dentro del dominio de soporte de la partícula j .

La representación integral de $f(\mathbf{x})$ puede escribirse ahora de forma discreta con la aproximación mediante partículas de la siguiente manera

$$\begin{aligned} f(\mathbf{x}) &= \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' \cong \sum_{j=1}^N f(\mathbf{x}_j)W(\mathbf{x} - \mathbf{x}_j, h)\Delta V_j \\ &= \sum_{j=1}^N f(\mathbf{x}_j)W(\mathbf{x} - \mathbf{x}_j, h)\frac{m_j}{\rho_j}. \end{aligned} \quad (17)$$

Finalmente, la aproximación mediante partículas de una función en una partícula i puede ser escrita como:

$$\langle f(\mathbf{x}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j)W_{ij}, \quad (18)$$

donde

$$W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h). \quad (19)$$

La ecuación (18) establece que el valor de una función en la posición de la partícula i es aproximada usando la suma de los valores de la función en todas las partículas que se encuentran dentro del dominio de soporte de la partícula i , ponderadas por la función kernel.

Siguiendo el mismo procedimiento, la aproximación mediante partículas de la derivada puede obtenerse como:

$$\langle \nabla \cdot f(\mathbf{x}) \rangle = - \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j) \cdot \nabla W(\mathbf{x}_i - \mathbf{x}_j, h), \quad (20)$$

donde el gradiente ∇W en la anterior ecuación es evaluado en la partícula j .

La ecuación (20) establece que el valor del gradiente de una función en una partícula localizada en la posición \mathbf{x} puede ser aproximada usando la suma de los valores de la función en la posición de todas las partículas dentro del soporte de dominio, multiplicado al volumen de la partícula y ponderado por el gradiente de la función kernel.

La aproximación mediante partículas convierte la forma continua de una función y sus derivadas en una suma discreta sobre un conjunto de partículas. El uso de estas sumas para la aproximación de la integral es un factor clave que hace el método SPH fácil de implementar.

Otro aspecto importante en el método es que se introduce la masa y la densidad de las partículas dentro de las ecuaciones. Este hecho puede ser convenientemente aplicado a los problemas hidrodinámicos donde la densidad es una variable clave. Para la aplicación a la mecánica de sólidos, se necesitaría un tratamiento especial.

2.2 Función kernel

Dentro del SPH es de máxima importancia determinar el patrón para las interpolaciones y definir la distancia del área de influencia de una partícula.

Numerosas investigaciones se han centrado en la función kernel, esperando mejorar el rendimiento del método SPH, y/o generalizar los requerimientos para construir la función kernel.

Las principales propiedades y condiciones que deben tener las funciones kernel, se pueden resumir en los siguientes puntos:

- La función kernel debe estar normalizada sobre el dominio de soporte. Condición de unidad de la ecuación (5).
- Las dimensiones del soporte compacto, ecuación (7), se define mediante el smoothing length h y un factor k que determina la extensión. Esta condición transforma la aproximación del SPH de una operación global a una local.
- $W(\mathbf{x} - \mathbf{x}') \geq 0$ para cualquier punto \mathbf{x}' dentro del dominio de soporte de la partícula \mathbf{x} . Esta propiedad establece que la función kernel debe ser no negativa dentro de dominio de soporte.
- El valor de la función kernel para una partícula debe ser monótonamente decreciente a medida que aumenta la distancia de la partícula. Esta propiedad está basada en la consideración física de que una partícula cercana debe tener una mayor influencia en la partícula bajo consideración. Es decir, la fuerza de interacción entre las partículas decrece con la distancia.
- La función kernel debe satisfacer la condición de la función delta de Dirac cuando el smoothing length se acerca a cero, ecuación (6).
- Para la aproximación de una función y sus derivadas, la función kernel necesita ser suficientemente continua para obtener buenos resultados.

Cualquier función que presente las anteriores propiedades puede ser empleada como función de interpolación en SPH.

En este trabajo se han implementado dos funciones kernel para la aproximación de las variables. La primera es Gaussian kernel utilizada por Gingold y Monaghan en sus artículos

$$W(\mathbf{x}, h) = \left(\frac{1}{h\sqrt{d}}\right)^d \exp(-\|\mathbf{x}\|^2/h^2). \quad (21)$$

Y por otro lado el cubic spline kernel:

$$W(\mathbf{x}, h) = C_h \begin{cases} (2-q)^3 - 4(1-q)^3 & 0 \leq q < 1 \\ (2-q)^3 & 1 \leq q < 2, \\ 0 & q \geq 2 \end{cases} \quad (22)$$

siendo d la dimensión del problema ($d = 1, 2, 3$)

$$q = \|\mathbf{x}\|/h \quad (23)$$

$$C_h = \begin{cases} \frac{1}{6h} & d = 1 \\ \frac{5}{14\pi h^2} & d = 2 \\ \frac{1}{4\pi h^3} & d = 3 \end{cases} \quad (24)$$

El gaussian kernel es la mejor elección para reproducir respuestas físicas (Gingold & Monaghan 1997) pero el spline kernel presenta la ventaja de ser distinto de cero únicamente en el soporte dominio, reduciendo así el tiempo de simulación.

Las magnitudes de ambos kernel y sus derivadas son representadas en la siguiente figura:

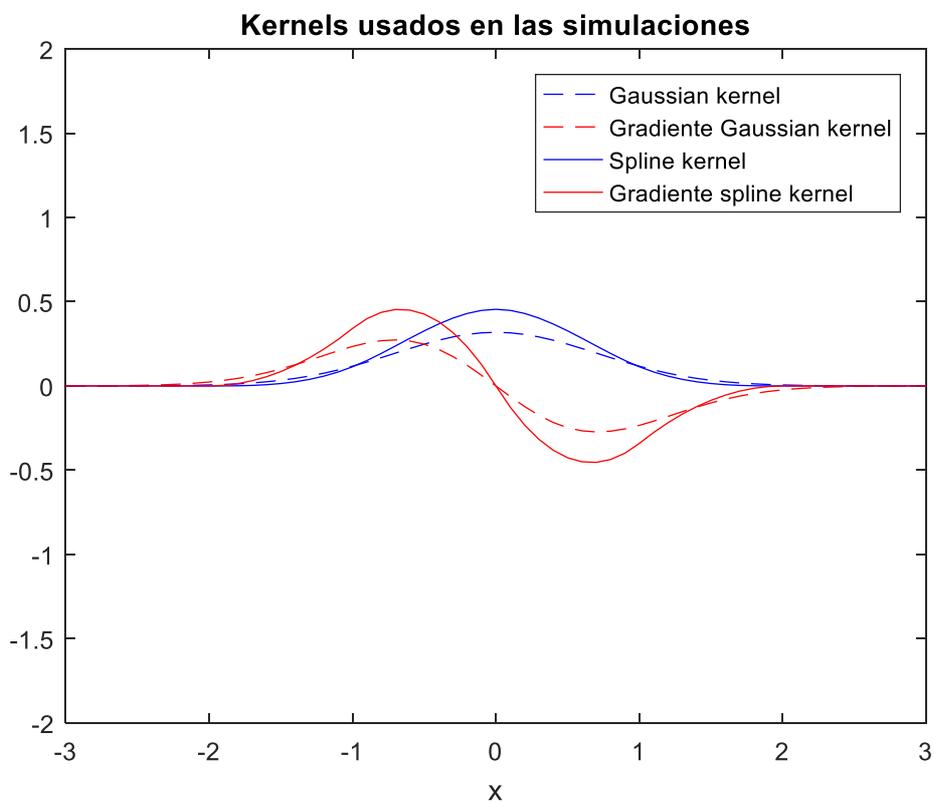


Figura (2-4)

2.3 Técnicas para obtener la formulación SPH

Mediante el uso de los procedimientos descritos hasta el momento, tales como la aproximación mediante kernel y partículas, la formulación SPH puede ser siempre obtenida. Sin embargo, existen numerosas posibilidades y caminos.

Monaghan emplea una aproximación directa mediante el uso de las ecuaciones (18) y (20). En su planteamiento se emplean las siguientes dos identidades para ubicar la densidad dentro del operador gradiente

$$\nabla \cdot f(\mathbf{x}) = \frac{1}{\rho} [\nabla \cdot (\rho f(\mathbf{x})) - f(\mathbf{x}) \cdot \nabla \rho], \quad (25)$$

$$\nabla \cdot f(\mathbf{x}) = \rho \left[\nabla \cdot \left(\frac{f(\mathbf{x})}{\rho} \right) + \frac{f(\mathbf{x})}{\rho^2} \cdot \nabla \rho \right]. \quad (26)$$

Estas dos identidades se pueden sustituir dentro de la integral (11). El mismo procedimiento de aproximación mediante partículas usado para obtener la ecuación (20) es aplicado a cada gradiente de las ecuaciones (25) y (26). Cada término fuera del gradiente se encuentra evaluado en la partícula bajo consideración. Los resultados sobre la divergencia de $f(\mathbf{x})$ en la partícula i son

$$\langle \nabla \cdot f(\mathbf{x}_i) \rangle = \frac{1}{\rho_i} \sum_{j=1}^N m_j [f(\mathbf{x}_j) - f(\mathbf{x}_i)] \cdot \nabla_i W_{ij}, \quad (27)$$

$$\langle \nabla \cdot f(\mathbf{x}_i) \rangle = \rho_i \left[\sum_{j=1}^N m_j \left[\frac{f(\mathbf{x}_j)}{\rho_j^2} + \frac{f(\mathbf{x}_j)}{\rho_j^2} \right] \cdot \nabla_i W_{ij} \right]. \quad (28)$$

Ambas expresiones tienen la ventaja de presentar una formulación simétrica o asimétrica, ayudando a mejorar la precisión numérica en las simulaciones SPH.

Además de las dos identidades ya mencionadas, otras reglas matemáticas son convenientes tener en cuenta a la hora de obtener la formulación SPH. Por ejemplo, para dos funciones arbitrarias f_1 y f_2 , tenemos la siguiente regla

$$\langle f_1 \pm f_2 \rangle = \langle f_1 \rangle \pm \langle f_2 \rangle, \quad (29)$$

$$\langle f_1 f_2 \rangle = \langle f_1 \rangle \langle f_2 \rangle. \quad (30)$$

La aproximación SPH de la suma de dos funciones es igual a la suma de las aproximaciones SPH de cada función individual, a la misma vez que la aproximación SPH del producto de dos funciones es igual a al producto de cada aproximación SPH.

Si f_1 es una constante denotada por c , se tendría:

$$\langle c f_2 \rangle = c \langle f_2 \rangle \quad (1.31)$$

2.4 Interpolación a partir de las partículas

El uso de la función kernel no se limita únicamente a la aproximación de las variables en las partículas. Una de las utilidades más interesantes es la interpolación de las variables en cualquier punto del espacio, a partir de los valores obtenidos en las partículas.

De esta manera podemos representar perfiles más completos. Para una función $f(x_k)$ siendo x_k un punto general del espacio, que puede ser la posición correspondiente a una partícula o no, el valor viene dado por

$$f(x_k) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) W_{kj}. \quad (1.32)$$

3 SPH APLICADO A DINÁMICA DE FLUIDOS

*“Las matemáticas son una ciencia exacta
salvo cuando te equivocas”
-Jaume Perich-*

Desde su invención en la década de 1970, SPH ha sido aplicado a numerosos problemas, incluyendo flujos compresibles e incompresibles, flujos con superficies libres, problemas de elasticidad y fractura... En este capítulo se detalla la formulación del método SPH y algunas aplicaciones interesantes para los problemas de dinámica de fluidos. La formulación deriva de discretizar las ecuaciones de Euler, llegando a un conjunto de ecuaciones diferenciales ordinarias con respecto al tiempo.

3.1 Ecuaciones de Euler en SPH

Las ecuaciones de Euler para la densidad ρ , velocidad \vec{v} y posición \vec{x} vienen dadas por

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \vec{v}, \quad (33)$$

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho} \nabla P, \quad (34)$$

$$\frac{d\vec{x}}{dt} = \vec{v}, \quad (35)$$

donde P denota la presión y las fuerzas gravitacionales han sido despreciadas. La primera ecuación también es llamada la ecuación de continuidad. En la ausencia de generación de entropía, se puede derivar de la primera ley de la termodinámica la variación de energía interna por unidad de masa u en cada instante como

$$\frac{du}{dt} = \frac{P}{\rho^2} \frac{d\rho}{dt}. \quad (36)$$

El conjunto de ecuaciones es completado con la ecuación de estado correspondiente. Para el caso de gases ideal se usará

$$P = \rho u (\gamma - 1), \quad (37)$$

siendo γ el coeficiente adiabático.

Las ecuaciones de continuidad, aceleración y energía se desarrollarán a continuación.

3.1.1 Ecuación continuidad

Usando los métodos de aproximación presentados en el capítulo 2, la ecuación de continuidad puede ser expresada como

$$\frac{d\rho_i}{dt} = \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij}. \quad (38)$$

Para determinados casos es ventajoso el uso de esta ecuación, sin embargo, para la gran mayoría de aplicaciones y simulaciones de SPH se emplea la siguiente ecuación

$$\rho_i = \sum_j m_j W_{ij}. \quad (39)$$

En ambos casos la conservación de la masa es exacta, debido a que las masas de las partículas generalmente son fijas en el método SPH.

3.1.2 Ecuación de la aceleración

Con el objetivo de asegurar la conservación del momento se utiliza la siguiente igualdad

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho. \quad (40)$$

La aproximación de la ecuación de la aceleración (32) se reescribe como

$$\frac{d\vec{v}_i}{dt} = - \sum_j m_j \left(\frac{P_j}{\rho_j^2} - \frac{P_i}{\rho_i^2} \right) \cdot \nabla_i W_{ij}. \quad (41)$$

Debido a la simetría de la ecuación, las fuerzas ahora son iguales y opuestas si se cumple la siguiente condición

$$\nabla_i W_{ij} = -\nabla_j W_{ji}. \quad (42)$$

Esta igualdad se cumple para valores iguales de smoothing length h en todas las partículas. Para el caso de que varíen no se cumpliría. Sin embargo, es posible mantener la conservación del momento variando h siempre y cuando se hagan simétricos los kernel. Una opción es usando la media aritmética de los smoothing lengths

$$\overline{h}_{ij} = \frac{1}{2}(h_i + h_j), \quad (43)$$

Y reescribiendo el kernel como:

$$\overline{W}_{ij} = W(\vec{x}_i - \vec{x}_j, \overline{h}_{ij}). \quad (44)$$

Con el objetivo de estabilizar el algoritmo se puede introducir viscosidad artificial a la ecuación, especialmente necesario cuando se simulan ondas de choque.

3.1.3 Viscosidad artificial

A la hora de simular los problemas hidrodinámicos se requiere un tratamiento especial para permitir que los algoritmos sean capaces de modelar las ondas de choque. La aplicación de la conservación de la masa, momento y energía a través de la onda de choque requiere la simulación de la transformación de energía cinética en energía térmica. Físicamente, esta transformación de energía puede representarse como disipación viscosa.

El modelo de viscosidad artificial desarrollado por Monaghan es el más usado en los algoritmos SPH. No solo proporciona la disipación necesaria para convertir la energía cinética en térmica, sino que también previene la penetración entre las partículas.

A continuación, se describe el modelo de viscosidad utilizado. Se añade un término Π_{ij} a la ecuación (39)

$$\frac{d\vec{v}_i}{dt} = - \sum_j m_j \left(\frac{P_j}{\rho_j^2} - \frac{P_i}{\rho_i^2} + \Pi_{ij} \right) \cdot \nabla_i W_{ij}, \quad (45)$$

donde Π_{ij} viene dado por:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & \vec{v}_{ij} \cdot \vec{r}_{ij} < 0 \\ 0 & \vec{v}_{ij} \cdot \vec{r}_{ij} > 0 \end{cases}. \quad (46)$$

Y la viscosidad

$$\mu_{ij} = \frac{\bar{h}_{ij} \vec{v}_{ij} \cdot \vec{r}_{ij}}{r_{ij}^2 + \eta^2}. \quad (47)$$

Con $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ y $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$. Las cantidades $\bar{\rho}_{ij}$ y \bar{c}_{ij} hacen referencia a la media de la densidad y la velocidad del sonido y son calculadas de la misma manera que el smoothing length en la ecuación (41).

Los parámetros α, β, η son normalmente: $\alpha = 1.0$, $\beta = 2.0$, $\eta = 0.1 \bar{h}_{ij}$. El término asociado a α representa la viscosidad de volumen, mientras el segundo término asociado con β intenta impedir la penetración entre partículas con altos números de Mach. Por otro lado, el factor η se introduce para prevenir divergencias numéricas cuando dos partículas se aproximan una a la otra.

3.1.4 Ecuación de la energía

La ecuación de la energía (34) se convierte en formulación SPH aproximando la ecuación de la densidad con la ecuación (36)

$$\frac{du_i}{dt} = \frac{P}{\rho_i^2} \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij}. \quad (48)$$

Si introducimos la viscosidad artificial (44), la ecuación de la energía se reescribe como:

$$\frac{du_i}{dt} = \frac{P}{\rho_i^2} \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij} + \frac{1}{2} \sum_j m_j \Pi_{ij} (\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij}.$$

3.2 Integración en el tiempo

En las matemáticas, la condición de Courant es necesaria para la convergencia en la resolución numérica de ciertos tipos de problemas. El diferencial de tiempo entre cada iteración ha de ser lo suficientemente pequeño para que la simulación no produzca resultados incorrectos.

Por ejemplo, en el caso de una onda moviéndose a través de una malla discreta si se quiere calcular la amplitud en periodos discretos de tiempo de igual duración, esta duración ha de ser menos que el tiempo que tarda la onda en viajar a la celda adyacente.

Para la elección del diferencial de tiempo en cada iteración, Monaghan propone un método para asegurar la estabilidad del algoritmo. Es una combinación de dos condiciones, una condición de fuerza

$$\Delta t_f = \min_i \sqrt{\frac{h}{|\vec{a}_i|}}, \quad (1.49)$$

junto con la condición de Courant, la cual como se ha visto incluye condiciones de estabilidad

$$\Delta t_c = \min_i \frac{h}{c_i + 0.6(\alpha c_i + \beta \max_j \mu_{ij})}. \quad (50)$$

La viscosidad μ_{ij} es dada por la ecuación (45) y los parámetros α, β son los mismos que en la ecuación (44).

El diferencial de tiempo es tomado por la siguiente expresión

$$\Delta t = 0.25 \min(\Delta t_f, \Delta t_c).$$

4 TUBO DE CHOQUE

“Si buscas resultados distintos, no hagas siempre lo mismo”

-Albert Einstein -

El problema de tubo de choque es una referencia muy utilizada para la evaluación de métodos numéricos. Muchos investigadores del SPH han estudiado este problema (Monaghan 1983, Hernquist 1989...), en el cual emplean el spline cúbico como función kernel.

4.1 Descripción del problema

El tubo de choque consiste en un tubo largo y recto relleno de gas, que es separado por una membrana en dos zonas con diferente densidad y presión. El gas en cada zona se encuentra en equilibrio con presión, densidad y temperatura constante.

Cuando la membrana es retirada, se produce una onda de choque, una onda de expansión o rarefacción en la zona de discontinuidad. La onda de choque se mueve hacia la región con menor densidad, mientras que la onda de expansión viaja a través de la zona con mayor densidad. La superficie de contacto se forma cerca del centro y se mueve dentro de la región de baja densidad detrás de la onda de choque.

En las siguientes figuras se pueden apreciar esquemas de la evolución del fluido

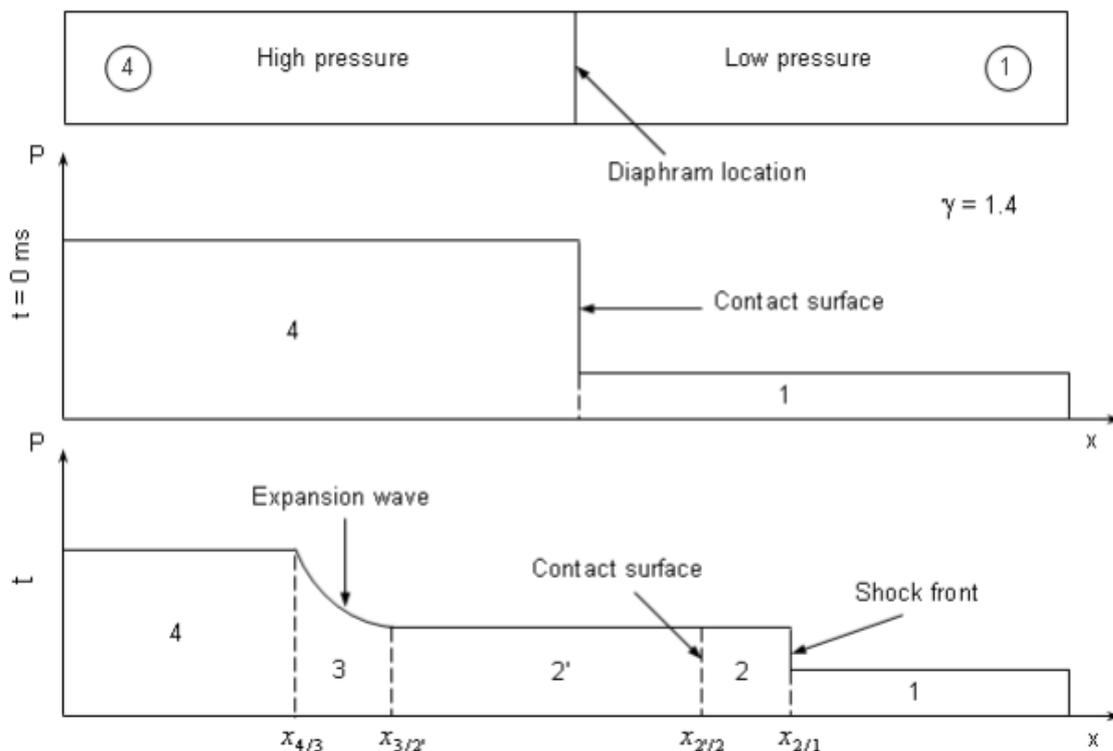


Figura (4-1)

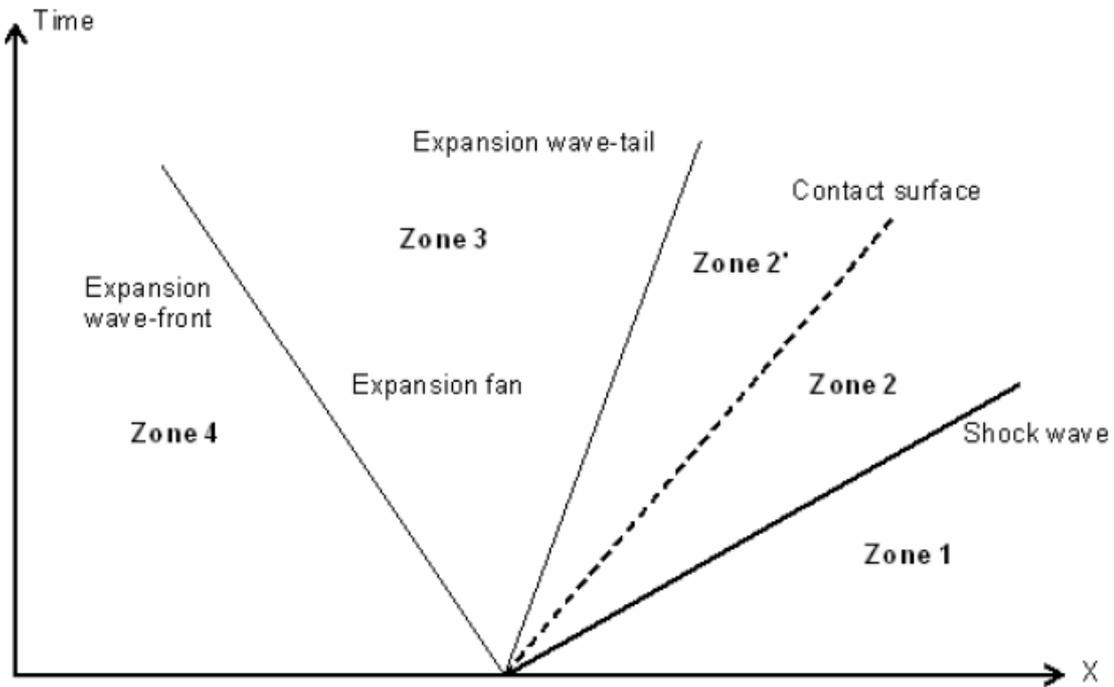


Figura (4-2)

Existen 4 zonas claramente diferenciadas como se puede ver en la figura (4-2).

La zona 1 corresponde al gas de baja presión que no ha sido modificado por la onda de choque. Dentro de la zona 2, distinguimos entre 2 y 2' por la superficie de contacto. Esta zona es la situada inmediatamente después de la onda de choque viajando a velocidad constante. A través de la superficie de contacto la densidad y la temperatura son discontinuas. La zona ubicada entre el inicio y el final de la onda de expansión corresponde a la zona 3, dentro de la cual las propiedades del fluido cambian gradualmente debido a que se trata de un proceso isentrópico. Por último, la zona 4 es la zona sin perturbar del gas a alta densidad.

4.2 Resultados

En esta simulación las condiciones iniciales vienen dado por el siguiente conjunto de valores

$$x \leq 0 \quad \rho = 1.0, P = 1.0, v = 0, u = 2.5, \quad (51)$$

$$x > 0 \quad \rho = 0.125, P = 0.1, v = 0, u = 2.0. \quad (52)$$

Para la simulación del problema se utilizan $N = 400$ partículas, donde $0.8 * N$ partículas se distribuyen uniformemente en la zona de alta densidad $x \in [-0.6, 0]$ mientras que el resto de partículas se ubican en la zona de baja densidad.

Esta distribución de partículas se debe a que la densidad es ocho veces mayor en la zona izquierda que en la derecha de la membrana, por consiguiente, el espaciado debe ser 8 veces menor en la zona de alta densidad.

La simulación del tubo de choque se resuelve mediante las ecuaciones de la densidad (37), aceleración (43) y la energía interna (47). La presión se obtiene mediante la ecuación de estado en cada intervalo de tiempo.

En la figura (4-3) se representa la evolución de la densidad vs la posición para el caso sin viscosidad artificial. Como era de esperar, se observan oscilaciones en la solución que tiende a ser inestable en la simulación de ondas de choque.

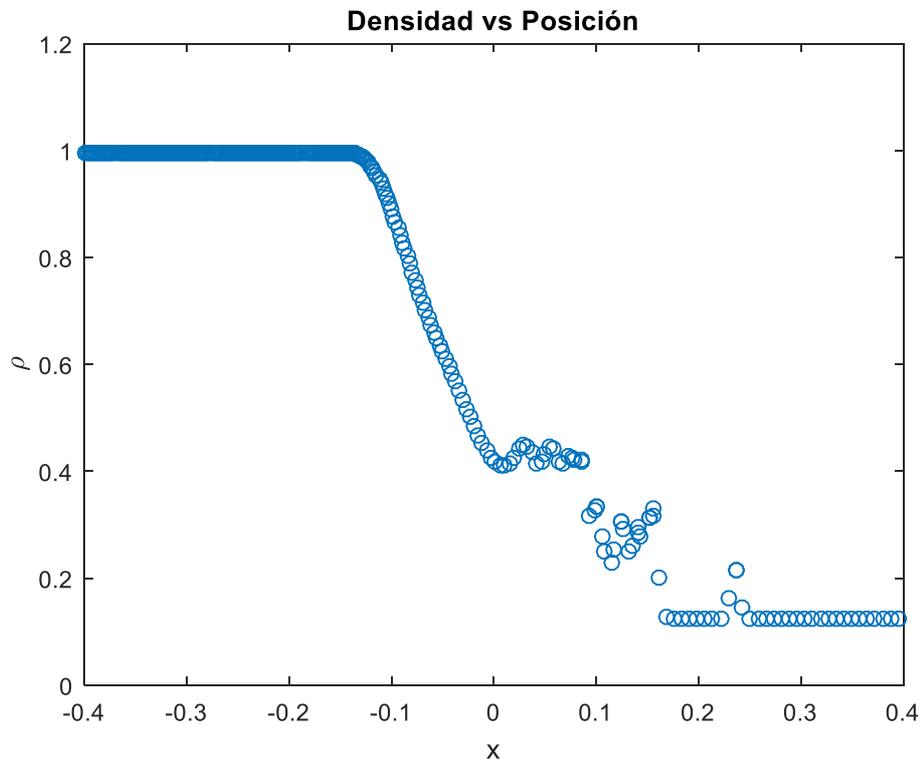


Figura (4-3)

Incluyendo los términos de viscosidad artificial se obtienen los siguientes perfiles:

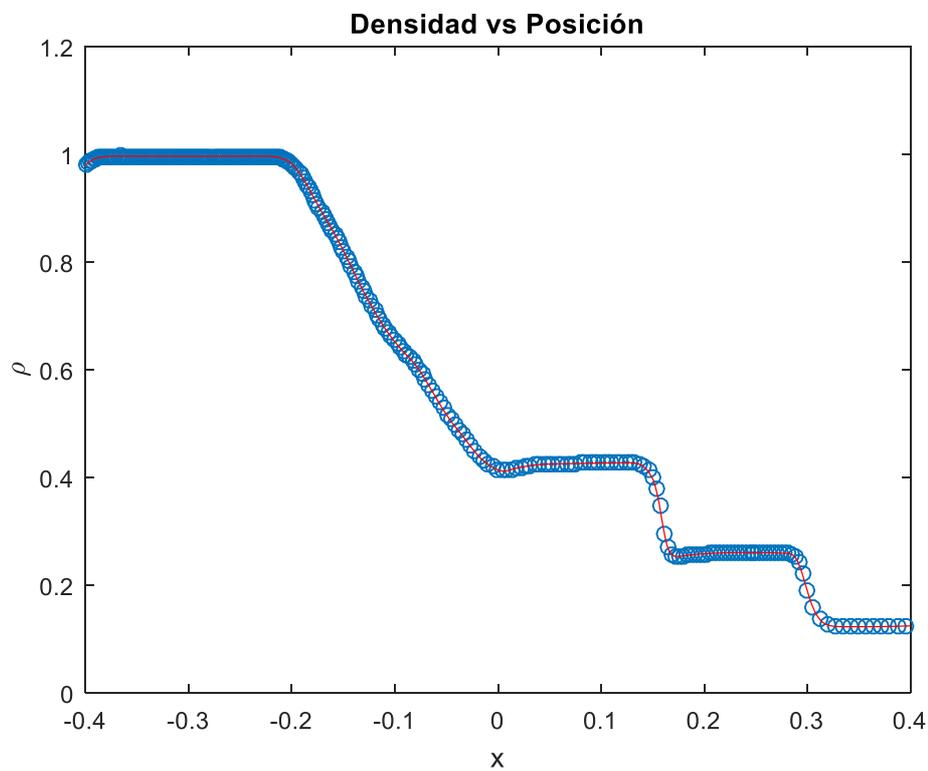


Figura (4-4)

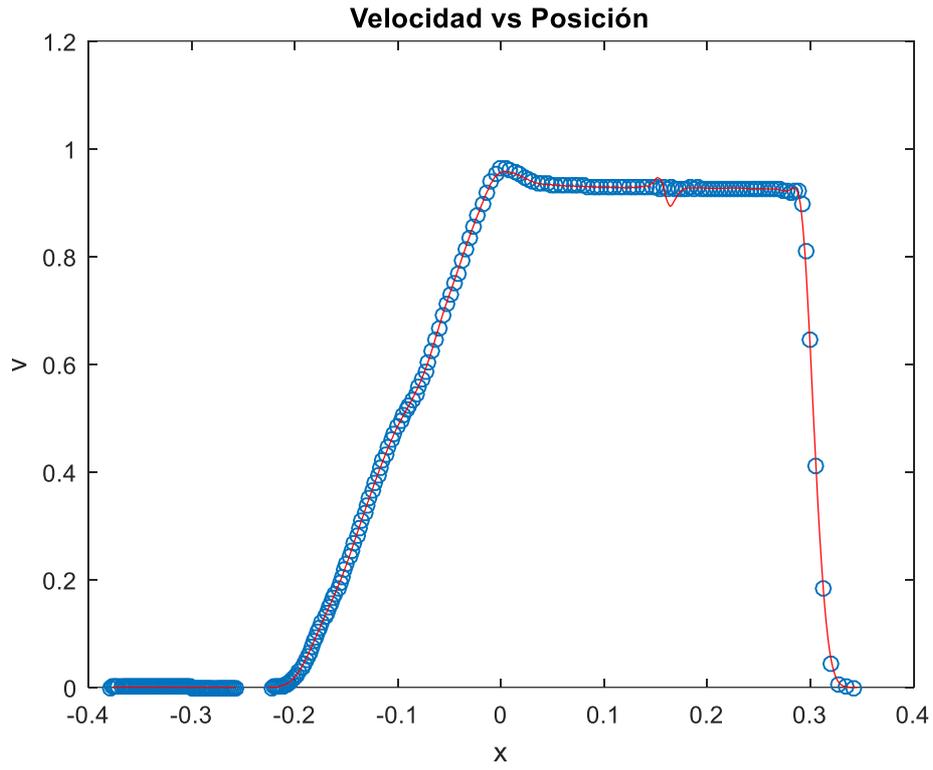


Figura (4-5)

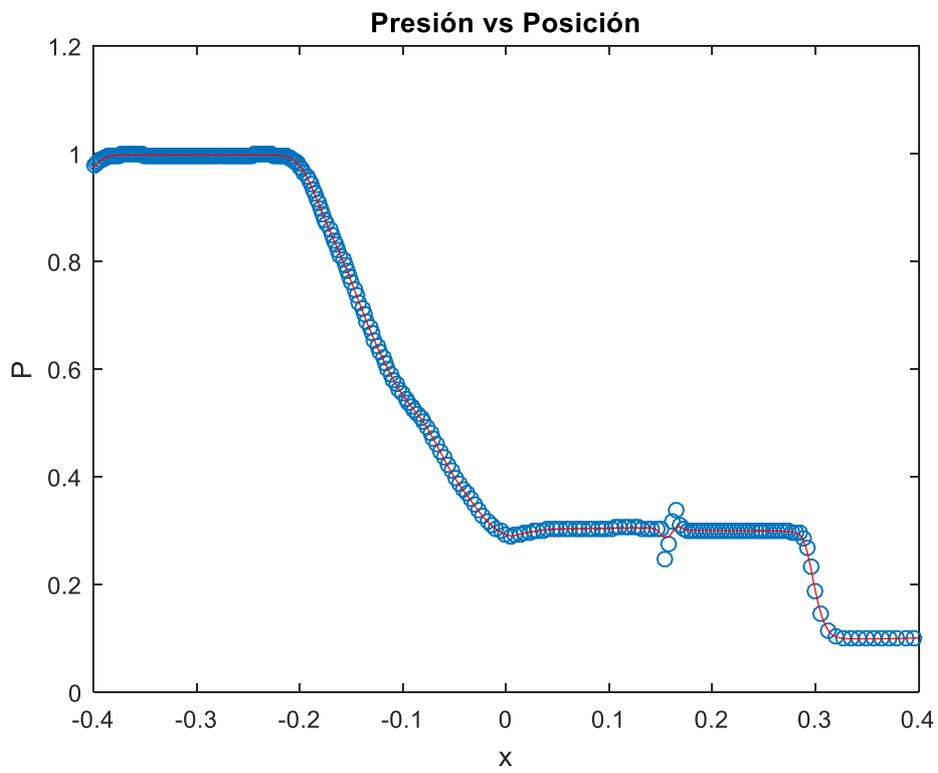


Figura (4-6)

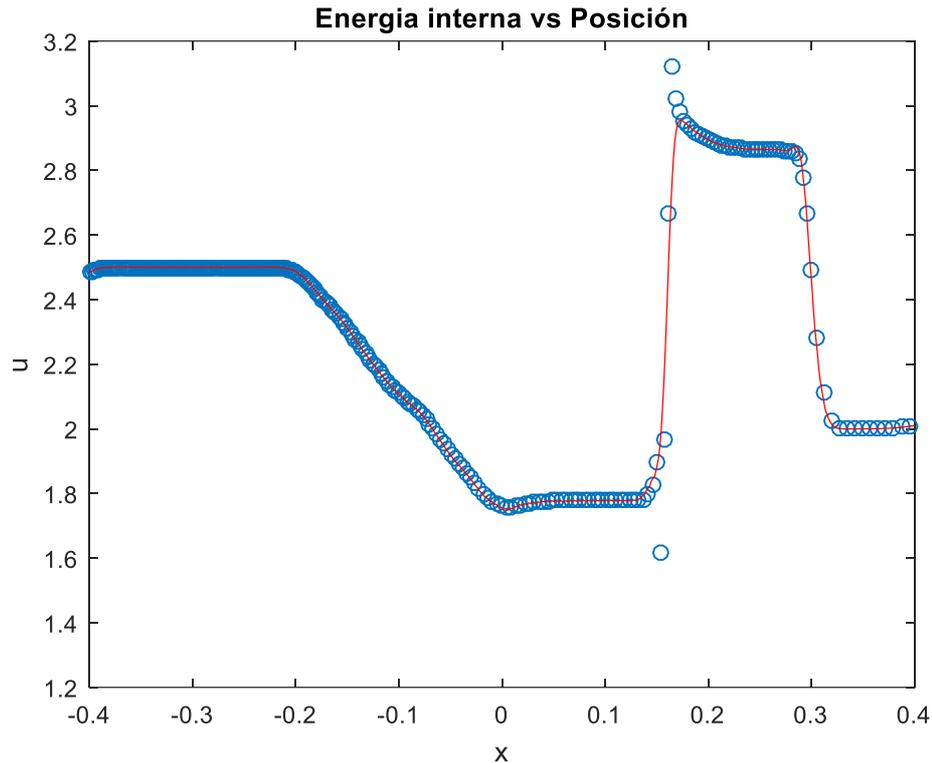


Figura (4-7)

4.3 Códigos MATLAB

Para acabar, se presentan los códigos empleados para la simulación del tubo de choque.

En los códigos existe una función principal donde se realiza la simulación y que contiene las ecuaciones de la aceleración, energía, densidad y presión incluyendo los términos viscosos. Dentro de esta función se llama a las funciones kernel empleadas, en este caso Spline kernel.

```
function shock_tube

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----FUNCIÓN PRINCIPAL TUBO DE CHOQUE-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
clc;
close all;

%% Datos y configuración inicial
Gam = 1.4;
N = 400; %Número de partículas
N1 = ceil(4*N/5);
N2 = N-N1;

%Zona de alta presión
x(1:N1,1) = linspace(-0.6,0,N1);
rho(1:N1,1) = 1;
m(1:N1,1) = 0.6*rho(1:N1,1)/N1;
v(1:N1,1) = 0;
u(1:N1,1) = 1/1/(gam-1);
```

```

%Zona de baja presión
x((N1+1):N,1) = linspace(0,0.6,N2);
rho((N1+1):N,1) = 0.125;
m((N1+1):N,1) = 0.6*rho((N1+1):N,1)/N2;
v((N1+1):N,1) = 0;
u((N1+1):N,1) = 0.1/0.125/(gam-1);

%Otros parámetros
H = 1*(x(N,1)-x(N-1,1));
Alpha = 1.0;
Beta = 2.0;
Eta = 0.1*h;

%% Integración en el tiempo

Ntime = 200; %Número máximo de iteraciones
T = 0;

for itime=1:Ntime
%Primero calculamos las distancias relativas y funciones kernel
for i=1:N
    for j=1:N,
        r(i,j) = x(i,1)-x(j,1);
        w(i,j) = spline_kernel(r(i,j),h,1);
        dw_dx(i,j) = spline_kernelp(r(i,j),h,1);
    end
end

% Calculo de la densidad

rho(:,1)=w*m;

% Calculo de la presion y velocidad del sonido

p(:,1)=(gam-1)*rho.*u;
c(:,1)=sqrt(gam*p./rho);

%Calculo del gradiente de presiones y viscosidad
for i=1:N,
    for j=1:N,

        f_p(i,j) = (p(i,1)/rho(i,1)^2+p(j,1)/rho(j,1)^2);
        cb(i,j) = 0.5*(c(i,1)+c(j,1));
        rhob(i,j)= 0.5*(rho(i,1)+rho(j,1));
        vdif(i,j)= v(i,1)-v(j,1);
        xdif(i,j)= x(i,1)-x(j,1);

        %viscosidad
        mu(i,j) = h*vdif(i,j)*xdif(i,j)/(xdif(i,j)^2+eta^2);
        Pi(i,j) = 0;

        if vdif(i,j)*xdif(i,j) < 0,

            Pi(i,j)=(-alpha*cb(i,j)*mu(i,j)+beta*mu(i,j)^2)/rhob(i,j);

        end
    end
end
end

```

```

aux1(:,1) = (vdif.*dw_dx)*m;
aux2(:,1) = 0.5*(Pi.*vdif.*dw_dx)*m;

%Condición de courant
Dtf      = min(sqrt(h./c));
mu_max(:,1) = max(mu')';
dtc      = min(h./(c+0.6*(alpha*c+beta*mu_max)));
dt       = 0.25*min(dtf,dtc);

%Actualización de las variables
T        = T+dt;
vt(:,1)  = v(:,1)-dt*((f_p+Pi).*dw_dx)*m;
ut(:,1)  = u(:,1)+dt*((p./rho.^2).*aux1+aux2);
xt       = x+0.5*dt*(vt+v);
rhot(:,1) = w*m;

% Promediado

Nr       = 1000;
r        = zeros(Nr,1);
r(:,1)  = linspace(-0.6,0.6,Nr);
rho_r   = zeros(Nr,1);

for i=1:Nr

    sum   = 0;
    sumv  = 0;
    sump  = 0;
    sumu  = 0;

    for j=1:N,
        uij = r(i,1)-x(j,1);
        sum = sum+m(j,1)*spline_kernel(uij,h,1);
        sumv = sumv+vt(j,1)*m(j,1)/rho(j,1)*spline_kernel(uij,h,1);
        sump = sump+p(j,1)*m(j,1)/rho(j,1)*spline_kernel(uij,h,1);
        sumu = sumu+ut(j,1)*m(j,1)/rho(j,1)*spline_kernel(uij,h,1);
    end

    rho_r(i,1)=sum;
    v_r(i,1)  =sumv;
    p_r(i,1)  =sump;
    u_r(i,1)  = sumu;
end

% Representacion
figure(1)
plot (xt,rhot,'o')
hold on
plot(r(:,1),rho_r,'r')
axis([-0.4 0.4 0 1.2])
pause(0.001)
title('Densidad vs Posición')
xlabel('x')
ylabel('\rho')
hold off

%Actualizando valores t-1
v=vt; u=ut; rho=rhot; x=xt;

end

```

```

figure(2)
plot(xt,vt, 'o')
hold on
plot(r(:,1),v_r, 'r')
axis([-0.4 0.4 0 1.2])
title('Velocidad vs Posición')
xlabel('x')
ylabel('v')

```

```

figure(3)
plot(xt,p, 'o')
hold on
plot(r(:,1),p_r, 'r')
axis([-0.4 0.4 0 1.2])
title('Presión vs Posición')
xlabel('x')
ylabel('P')

```

```

figure(4)
plot(xt,ut, 'o')
hold on
plot(r(:,1),u_r, 'r')
axis([-0.4 0.4 1.2 3.2])
title('Energia interna vs Posición')
xlabel('x')
ylabel('u')
%
%
```

```
function w=spline_kernel(r,h,d)
```

```

%SPLINE_KERNEL- Smoothing kernel function, reduces computation time.
%               Format:w=gaussian_kernel(r,h,d)
%               d-dimension of the problem
%               h-kernel length

```

```

q = norm(r)/h;
ch = [1/(h), 15/(7*pi*h^2), 3/(2*h^3*pi)];

```

```
if 0<=q && q<1
```

```
    w = 2/3-q^2+q^3/2;
```

```
elseif 1<=q && q<2
```

```
    w = 1/6*(2-q)^3;
```

```
else
```

```
    w = 0;
```

```
end
```

```
w = ch(d)*w;
```

```

function dw_dr=spline_kernelp(r,h,d)
%SPLINE_KERNELP- Gradient smoothing kernel function, reduces computation
time.
%           Format:w=gaussian_kernelp(r,h,d)
%           d-dimension of the problem
%           h-kernel length

q = norm(r)/h;
ch = [1/(h),15/(7*pi*h^2),3/(2*h^3*pi)];

if 0<=q && q<1

    dw_dr = -2*r/(h^2)+3/4*norm(r)*2*r/h^3;

elseif 1<=q && q<2

    dw_dr = 1/2*(2-q)^2*(-r/(h*norm(r)));

else

    dw_dr = 0;

end

dw_dr = ch(d)*dw_dr;

```

5 MODELO SIMPLIFICADO DE ESTRELLA

“Debemos intentar comprender el comienzo del universo a partir de bases científicas. Puede que sea una tarea más allá de nuestras capacidades, pero al menos deberíamos intentarlo”

-Stephen Hawking-

En este capítulo se desarrolla la aplicación del método SPH a modelos de estrellas politrópicas. Primero se presenta el modelo físico simplificado de una estrella. Dicho modelo se basa en los artículos de Monaghan & Price (2004) y trabajos posteriores. Las condiciones iniciales escogidas se basan en el artículo [3] influenciados por los trabajos realizados por Braune & Lewiner (2009).

Finalmente se presentan los resultados de las diversas simulaciones llevadas a cabo. Se comparan los resultados en los estados de equilibrio con la solución analítica. En las simulaciones se emplean diversas funciones kernel, smoothing length ... analizando cómo afecta su variación a la simulación.

5.1 Descripción del problema

El modelo simplificado de estrella de Monaghan & Price (2004) viene descrito por la siguiente ecuación

$$\frac{d\mathbf{v}_i}{dt} = -\nu\mathbf{v}_i - \sum_{j,j \neq i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) - \lambda\mathbf{x}_i, \quad (53)$$

donde ν es el coeficiente de amortiguación añadido para que la simulación alcance el estado de equilibrio. El término λ es una simplificación del término gravitacional. Para el caso estático ($\mathbf{v} = 0$) se puede resolver analíticamente la ecuación diferencial

$$0 = -\frac{1}{\rho} \nabla P - \lambda\mathbf{x} = -\frac{k(1 + 1/n)}{\rho} \rho^{1/n} \nabla \rho - \lambda\mathbf{x}. \quad (54)$$

Obteniendo que la densidad viene determinada por el siguiente perfil

$$\rho(r) = \left(\frac{\lambda}{2k(1+n)} (R^2 - r^2) \right)^n. \quad (55)$$

donde r es la posición radial y R es el radio de la estrella, más allá del radio de la estrella la densidad es cero.

Para el caso de $n = 1$, tenemos el siguiente perfil de densidad

$$\rho(r) = \frac{\lambda}{4k} (R^2 - r^2). \quad (56)$$

Si le asignamos a la estrella un radio R y una masa M , entonces la integral de la densidad ha de ser igual a la masa de la estrella

$$M = \int d^d r \rho(r). \quad (57)$$

siendo d la dimensión del problema. Esta integral nos permite obtener una solución para el parámetro λ :

$$\lambda = \begin{cases} \frac{2k\pi^{-1/n} \left(\frac{M(1+n)}{R^2}\right)^{1+\frac{1}{n}}}{M} & d = 2 \\ \frac{2k(1+n)\pi^{-3/2n} \left(\frac{M\Gamma\left(\frac{5}{2}+n\right)}{R^3\Gamma(1+n)}\right)^{\frac{1}{n}}}{R^2} & d = 3 \end{cases}. \quad (58)$$

5.2 Resultados

Las condiciones iniciales para la resolución numérica de las ecuaciones son los siguientes:

PARÁMETRO	VALOR
NÚMERO DE PARTICULAS	$N = 100..$
DIMENSIÓN	$d = 2.$
MASA DE LA ESTRELLA	$M = 2.$
RADIO DE LA ESTRELLA	$R = 0.75.$
SMOOTHING LENGTH	$h = 0.04/\sqrt{N/1000}.$
TIEMPO DIFERENCIAL	$dt = 0.04.$
VISCOCIDAD	$\nu = 1.$
CONSTANTE DE PRESIÓN	$k = 0.1.$
ÍNDICE POLITRÓPICO	$n = 1.$
MÁXIMO NÚMERO DE ITERACIONES	400.
KERNEL	Spline

La configuración inicial corresponde a una distribución aleatoria dentro de un círculo de radio R.

Las simulaciones son llevadas a cabo hasta alcanzar las condiciones de equilibrio, tiempo que se determina monitorizando la densidad central de la estrella.

En primer lugar, se prueba el efecto de la variación del número de partículas empleadas. Variando N entre 50, 100 y 300 partículas. Se representa la posición de cada partícula con su densidad para tres instantes de tiempo:

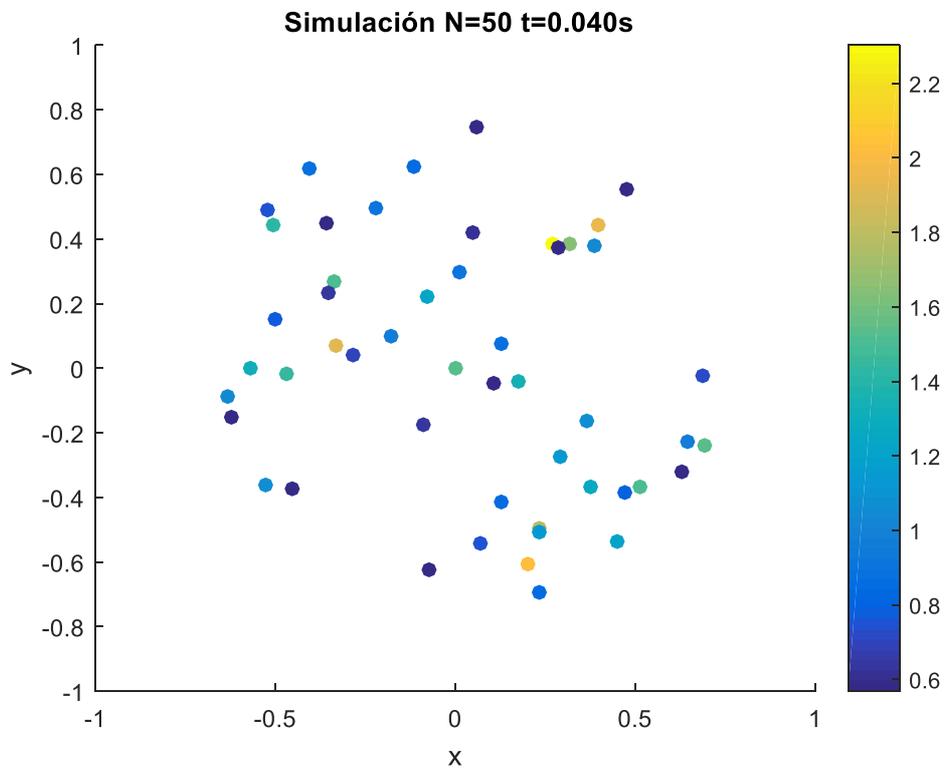


Figura (5-1)

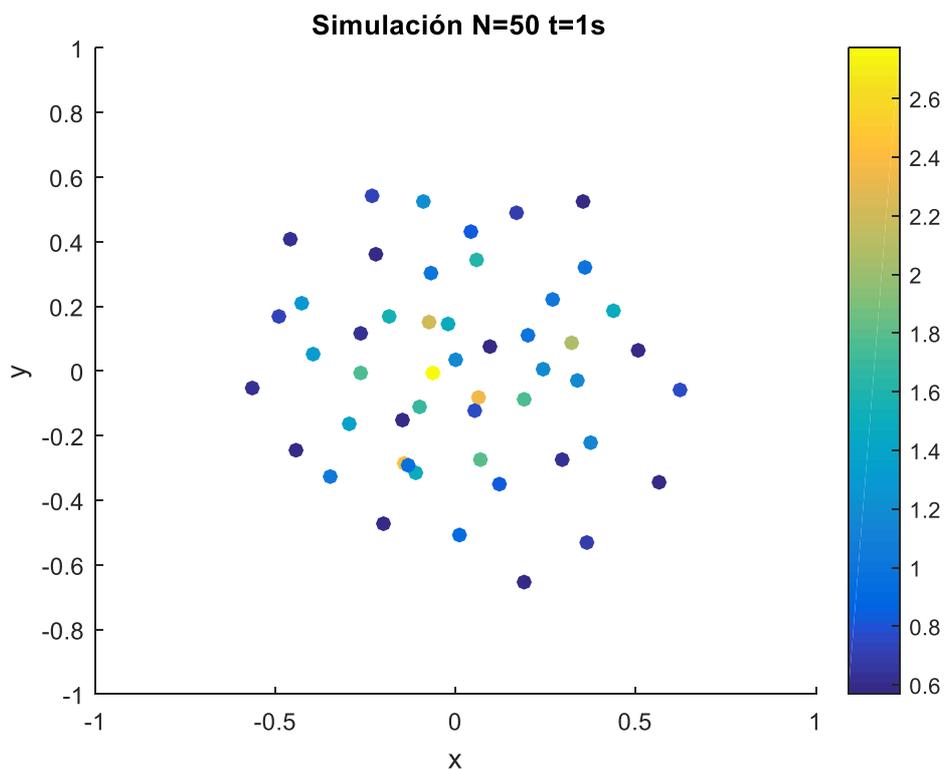


Figura (5-2)

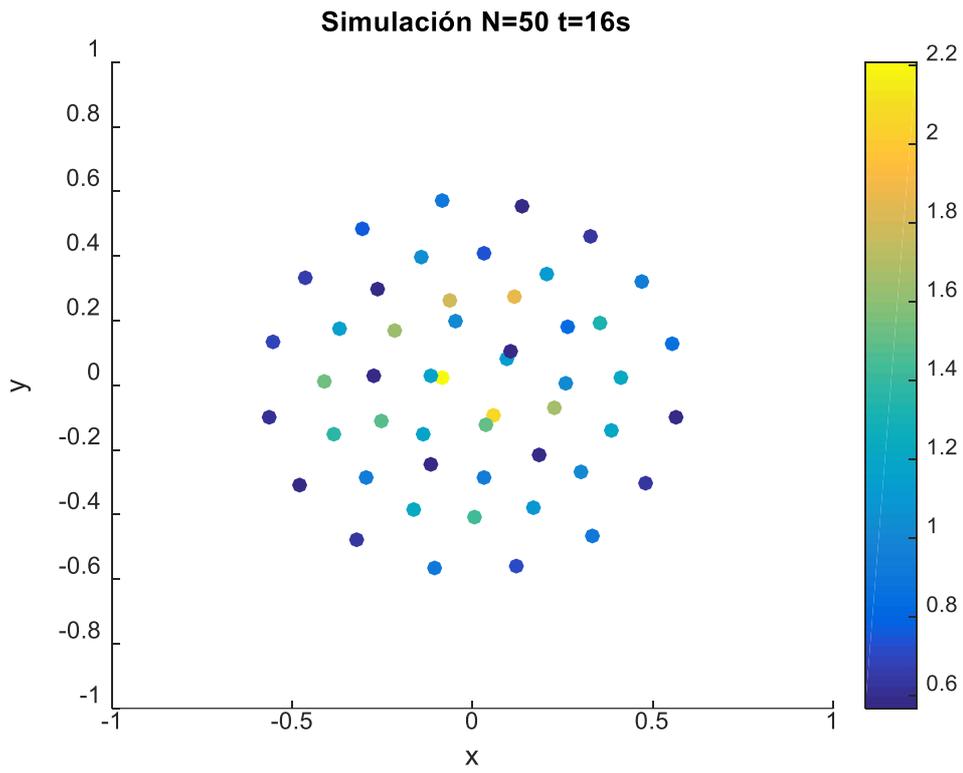


Figura (5-3)

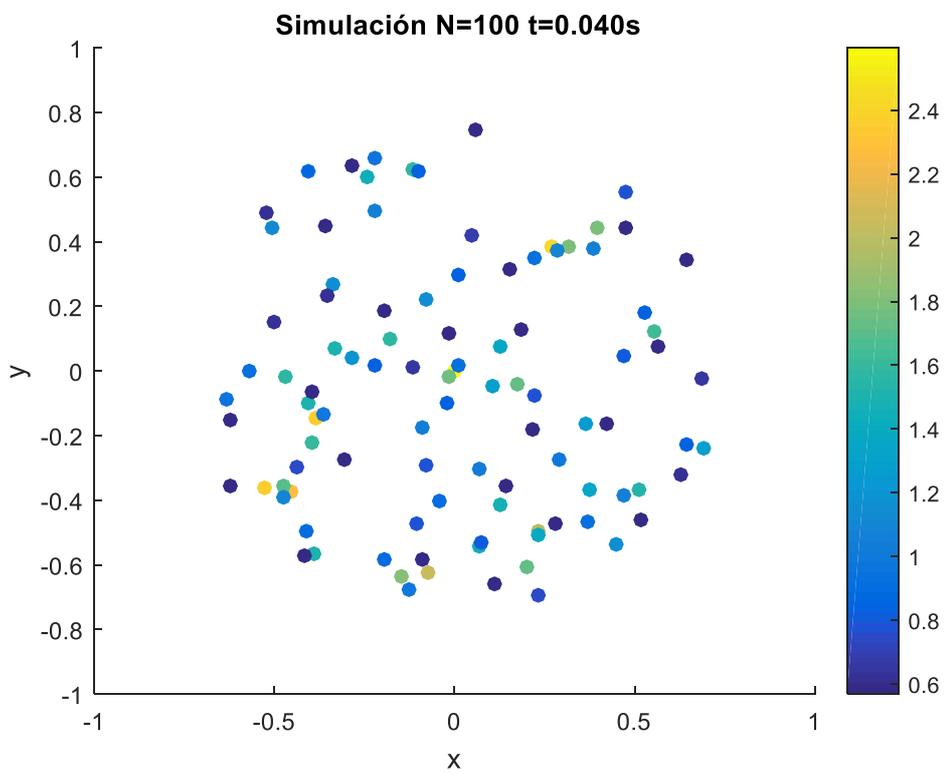


Figura (5-4)

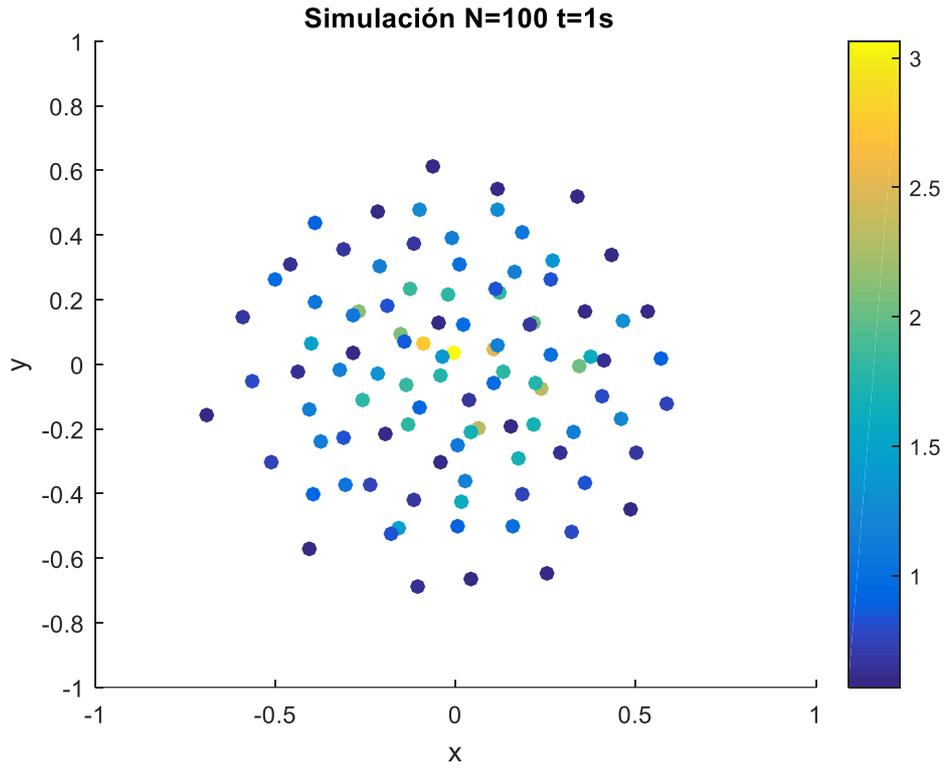


Figura (5-5)

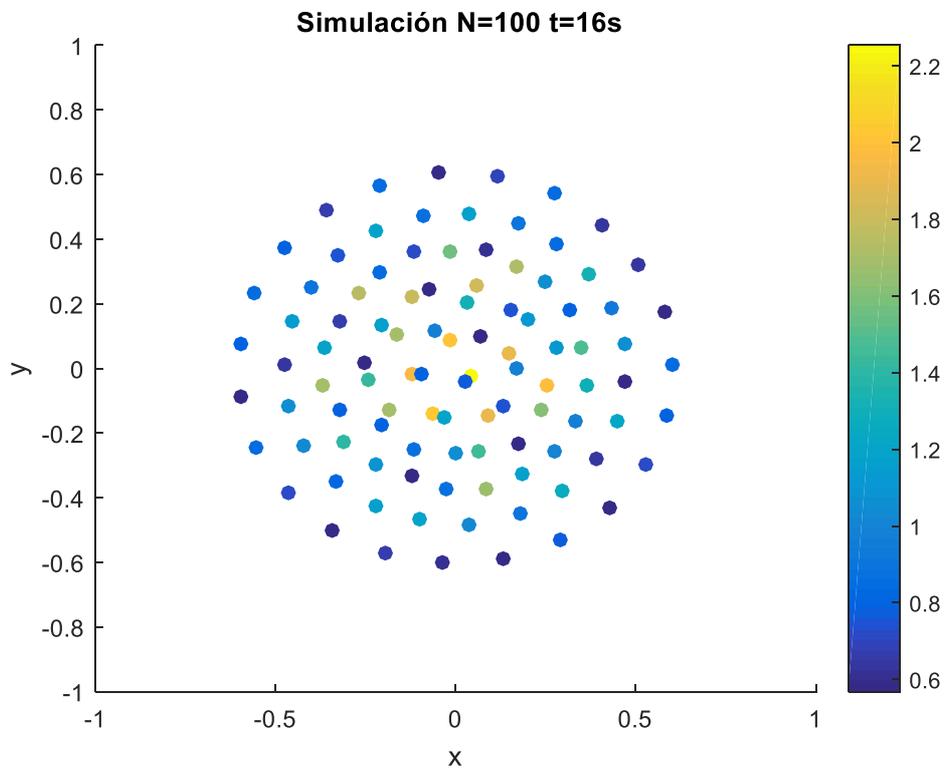


Figura (5-6)

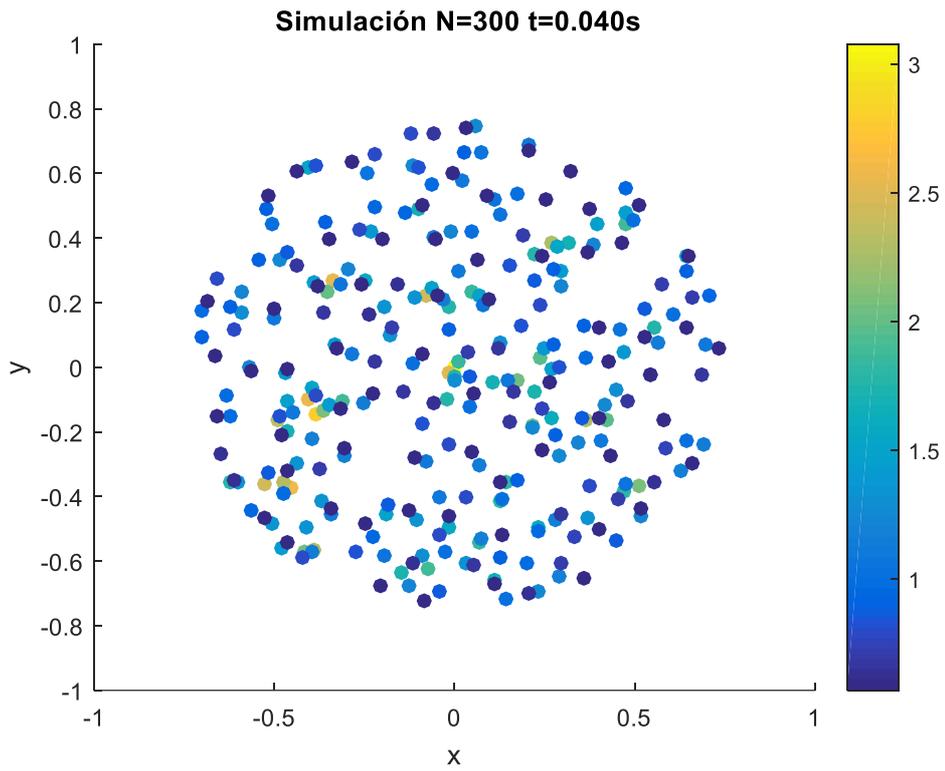


Figura (5-7)

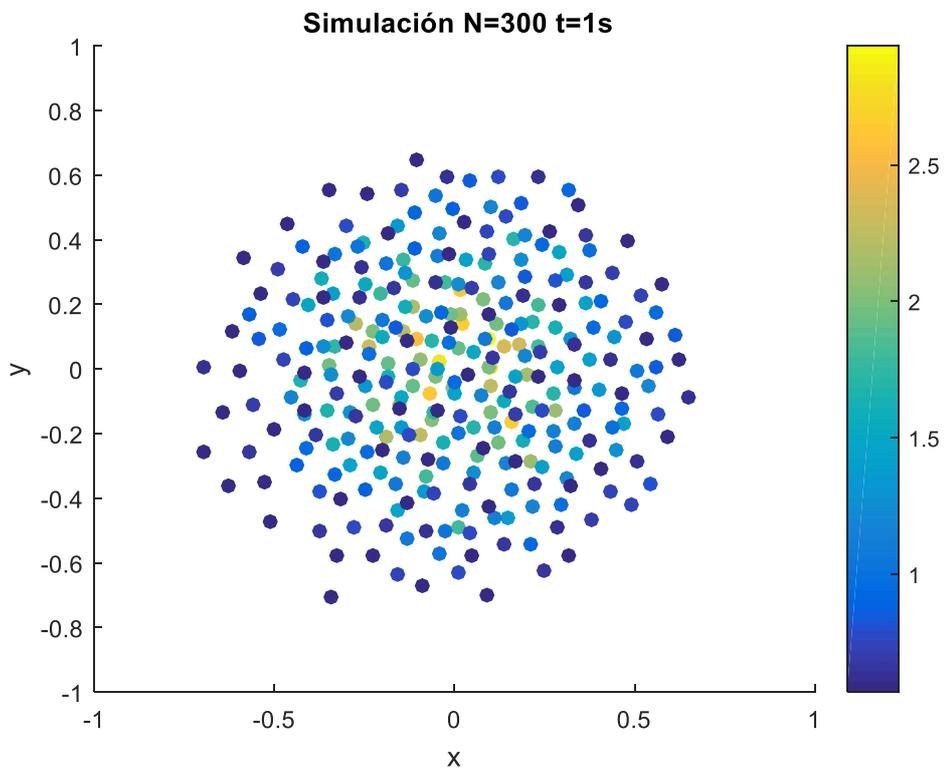


Figura (5-8)

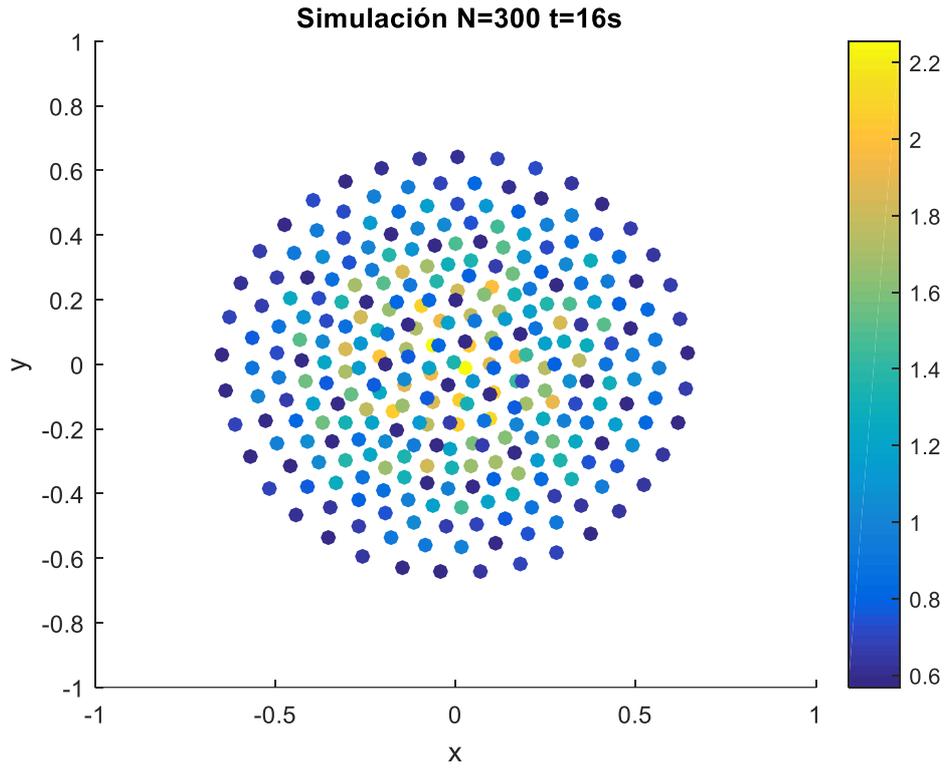


Figura (5-9)

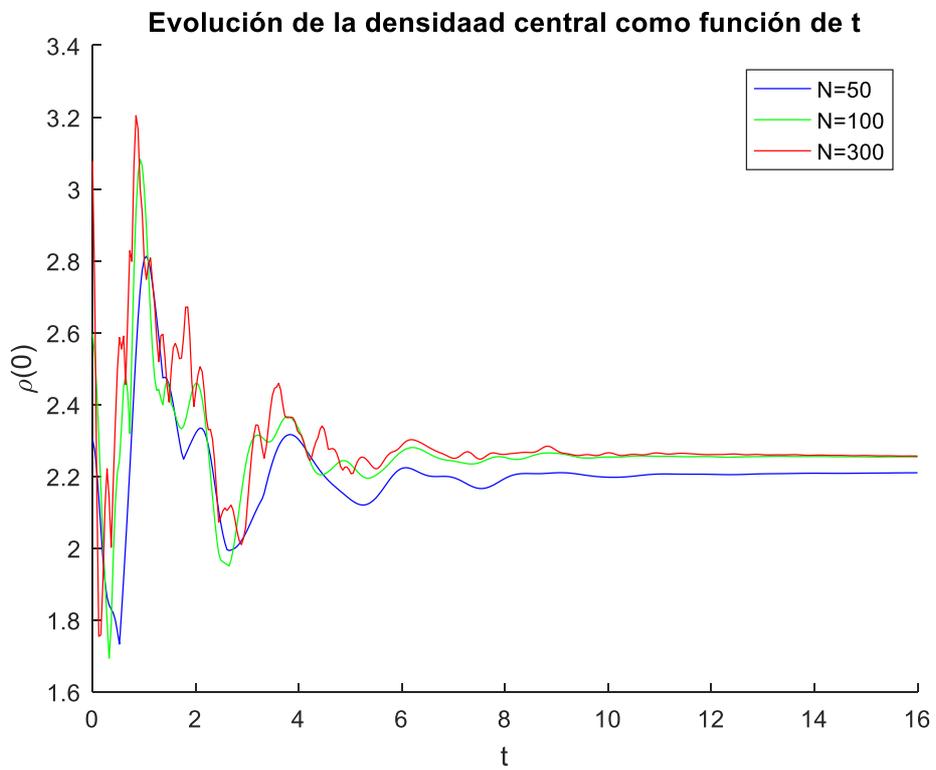


Figura (5-10)

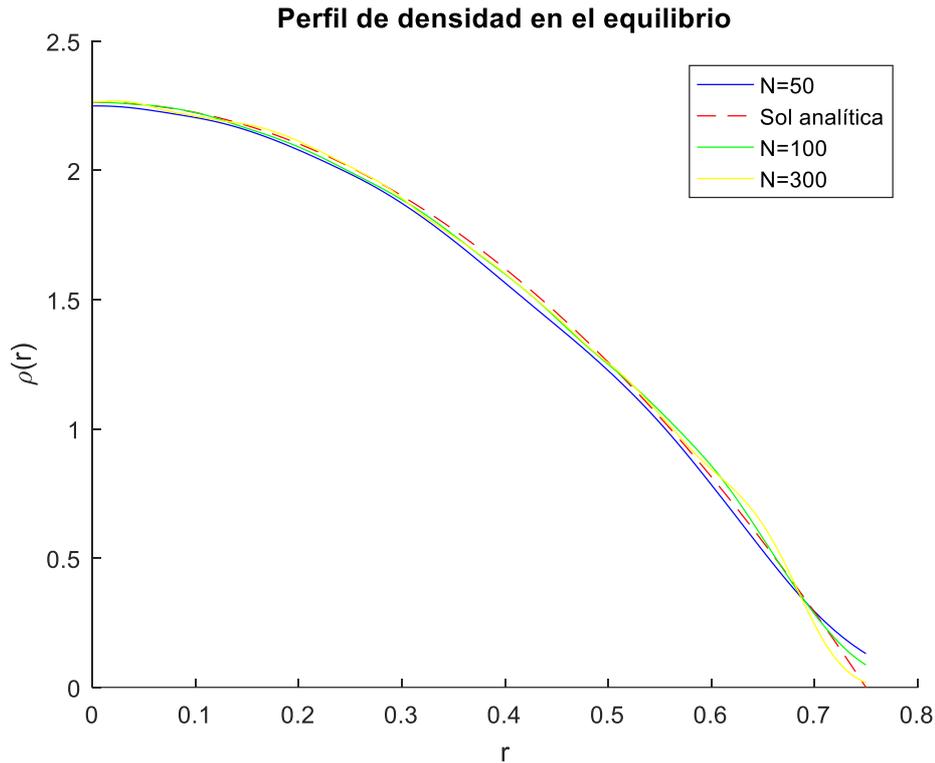


Figura (5-11)

De las primeras simulaciones se observa una serie de aspectos claves del método SPH en su comparación con los resultados analíticos. Ante todo, la reproducción de la solución analítica es bastante buena.

La densidad central $\rho(0)$ se aproxima mejor cuanto mayor sea el número de partículas N empleado. Usando únicamente $N = 50$ partículas la densidad obtenida es ligeramente inferior a la obtenida por la expresión analítica. Para corregir este hecho, se emplean más partículas y se puede observar que el valor obtenido con $N = 100$ y $N = 300$ es muy cercano al valor buscado.

Otro detalle a comentar es cómo en la representación del perfil de densidades, la aproximación se desvía de la solución analítica cuando $r \rightarrow R$. En el modelo analítico la densidad converge a cero. En este caso la forma de la función kernel altera el perfil de densidad siendo algo difícilmente corregible en el método SPH. Variando el smoothing length podemos llegar a mejores resultados.

Generalmente, una reducción de h proporciona una mejora en la respuesta siempre y cuando existan suficientes partículas en los alrededores de cada partícula. Reducir h para obtener mejores respuestas implica aumentar el número de partículas.

A continuación, se varía las funciones kernel presentadas en el trabajo, analizando su efecto en la evolución de la densidad central y el perfil de densidad. Cambiando entre Gaussian kernel y Spline kernel se obtienen las siguientes gráficas:

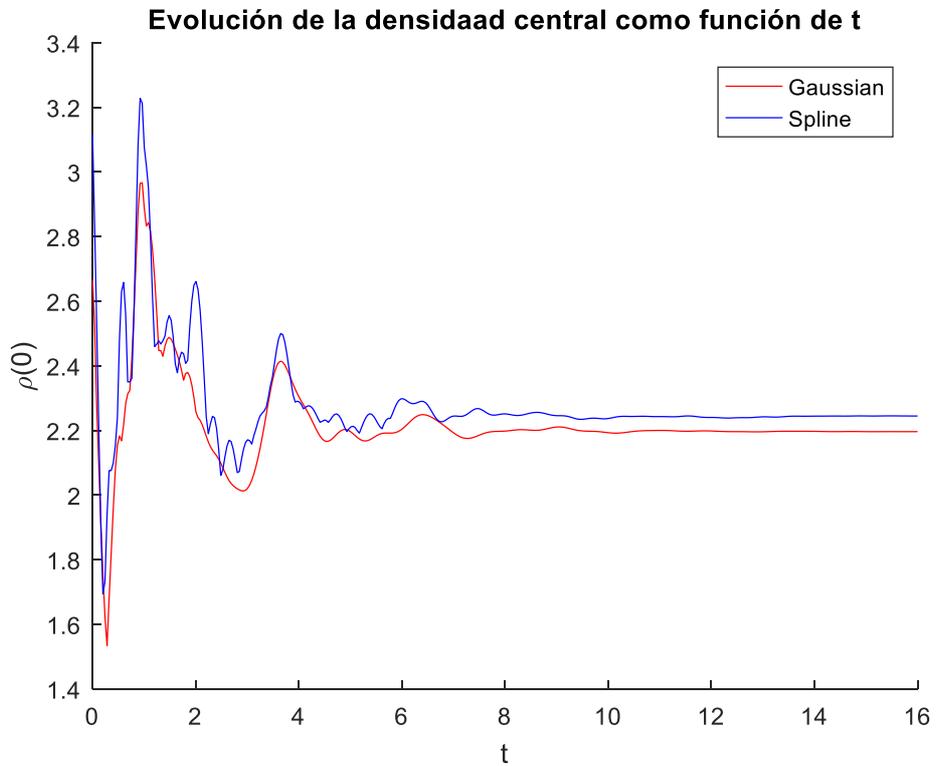


Figura (5-12)

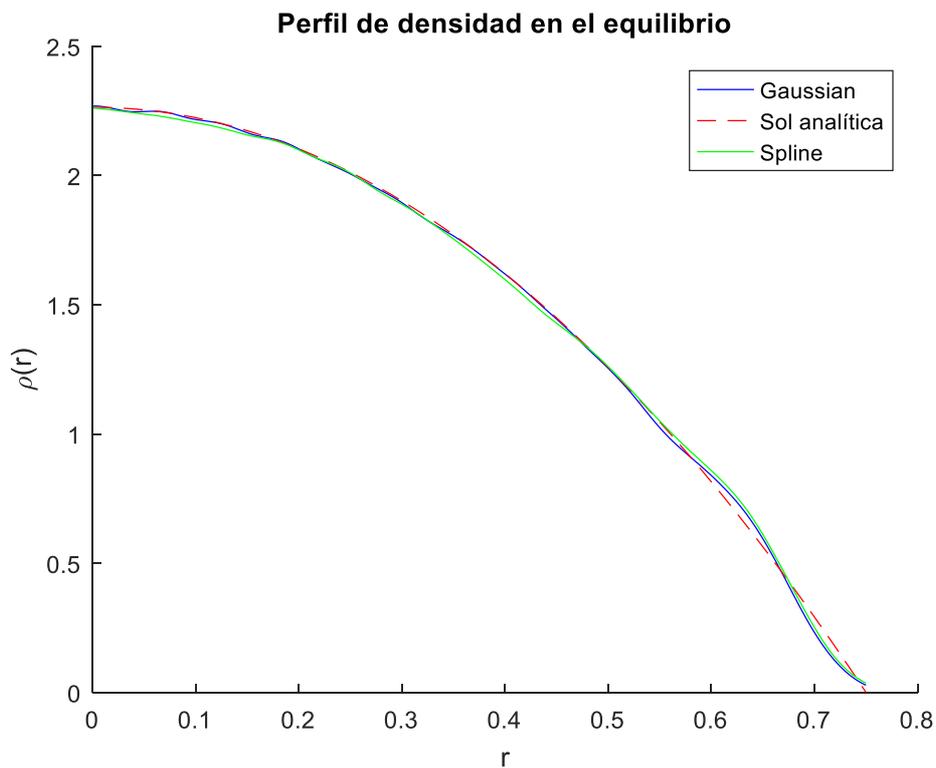


Figura (5-13)

Aparentemente no se aprecia variación significativa entre la utilización del Gaussian kernel o el spline kernel en el perfil de densidad, como podemos ver en la Figura (5-13).

Gingold & Monaghan recomiendan la utilización del Gaussian kernel para obtener mejores respuestas físicas, sin embargo, el spline kernel también es efectivo y realiza los cálculos más eficientemente como ya se ha explicado.

Por último, se varía el valor empleado de la amortiguación:

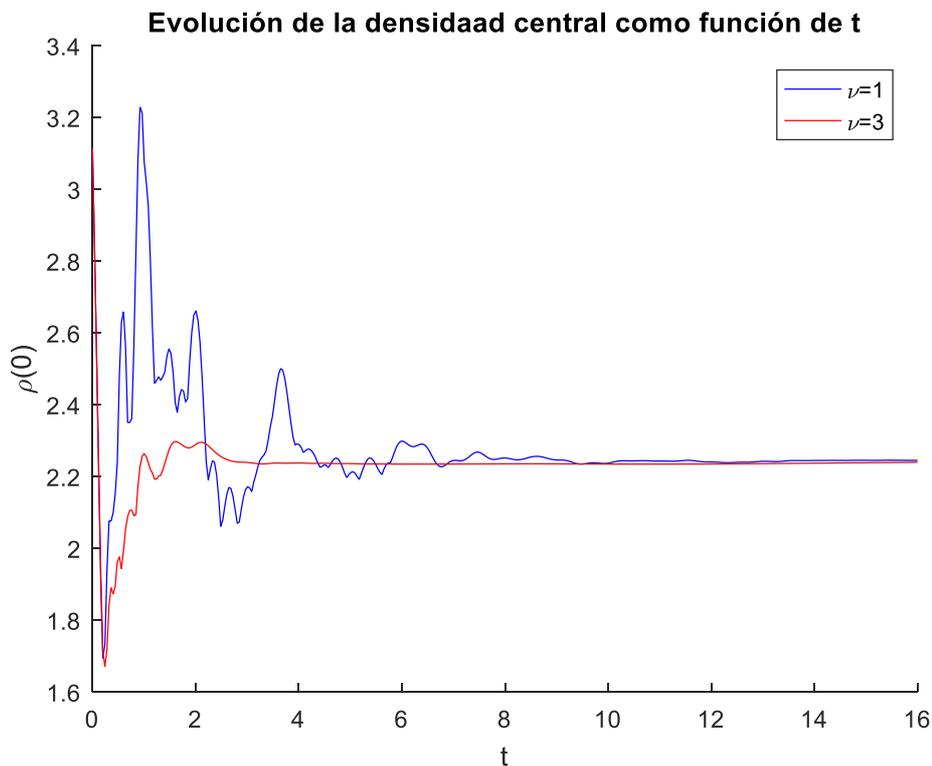


Figura (5-14)

El incremento de ν reduce el tiempo que emplea la estrella en llegar al equilibrio, Figura (5-14). Sin amortiguación la estrella continuaría oscilando indefinidamente.

Además, también se ha simulado la colisión de dos estrellas con una velocidad relativa de $v = 0.4$, representando la posición de las partículas y el valor de la presión y densidad máximas como función del tiempo en las figuras (5-18) y (5-19). Ejemplo claro de como el método SPH puede ser usado para monitorizar la evolución de parámetros físicos en el tiempo que de otra manera serían muy difícil o imposible de calcular analíticamente.

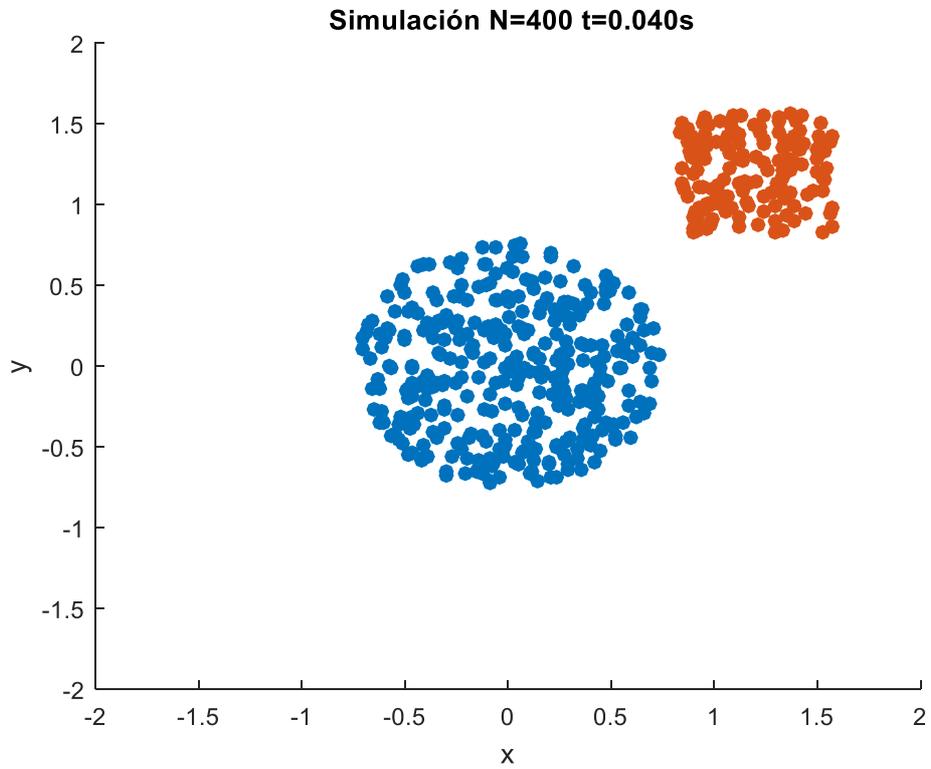


Figura (5-15)

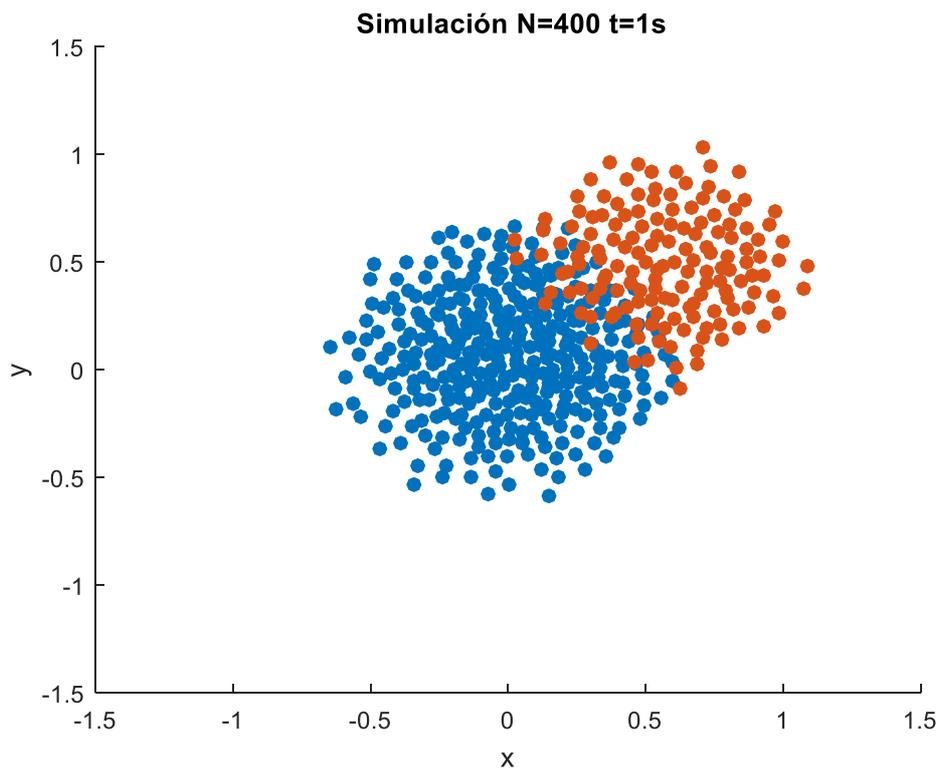


Figura (5-16)

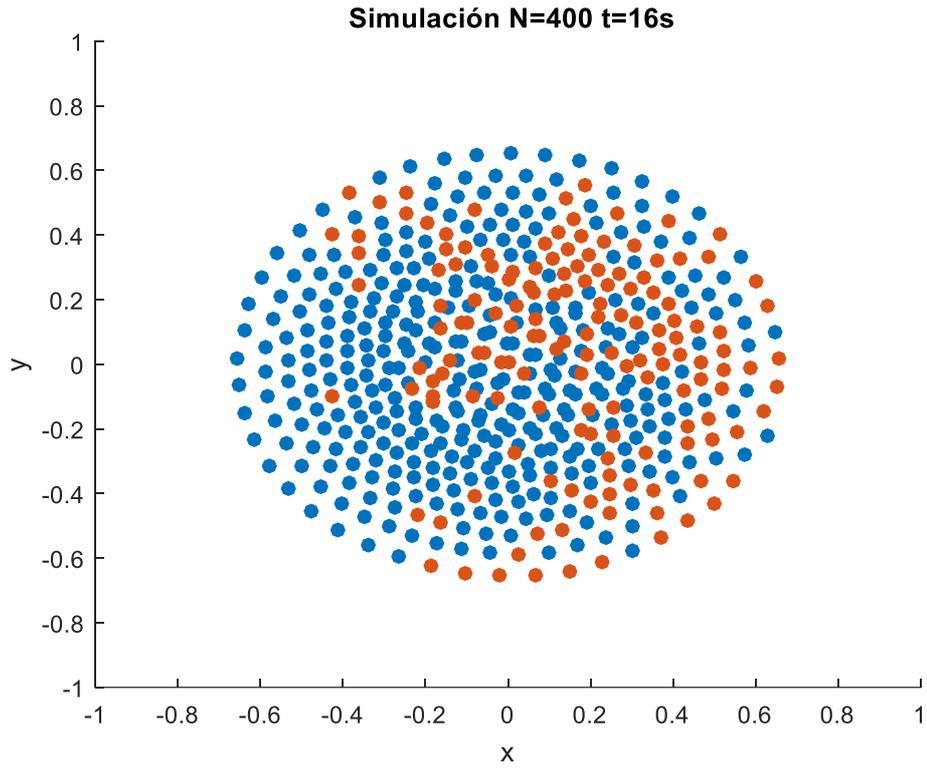


Figura (5-17)

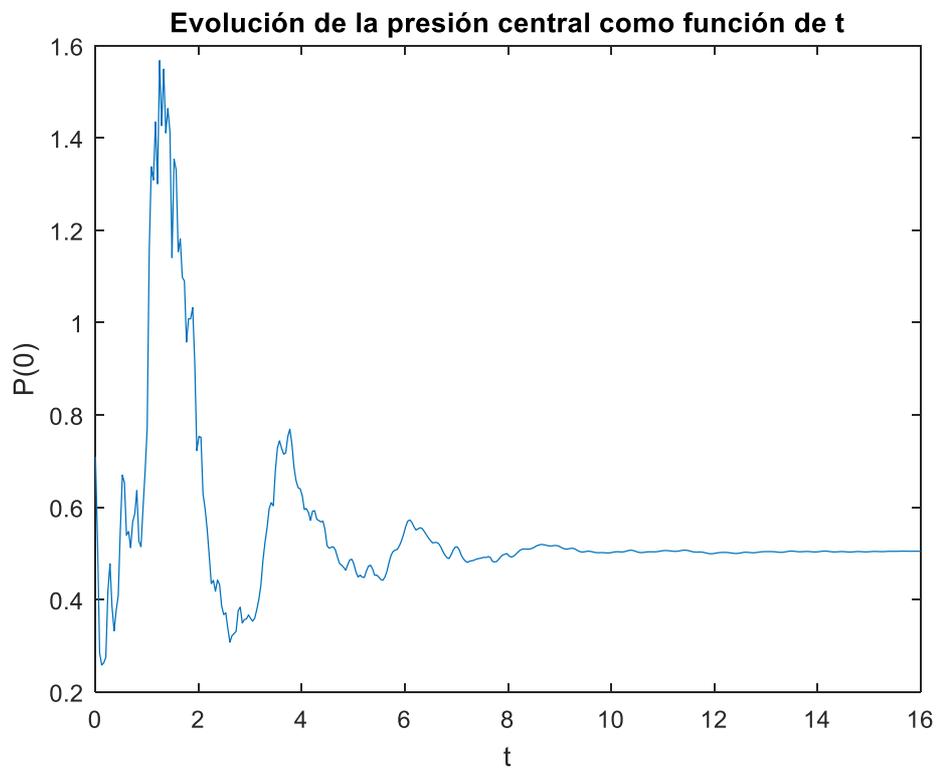


Figura (5-18)

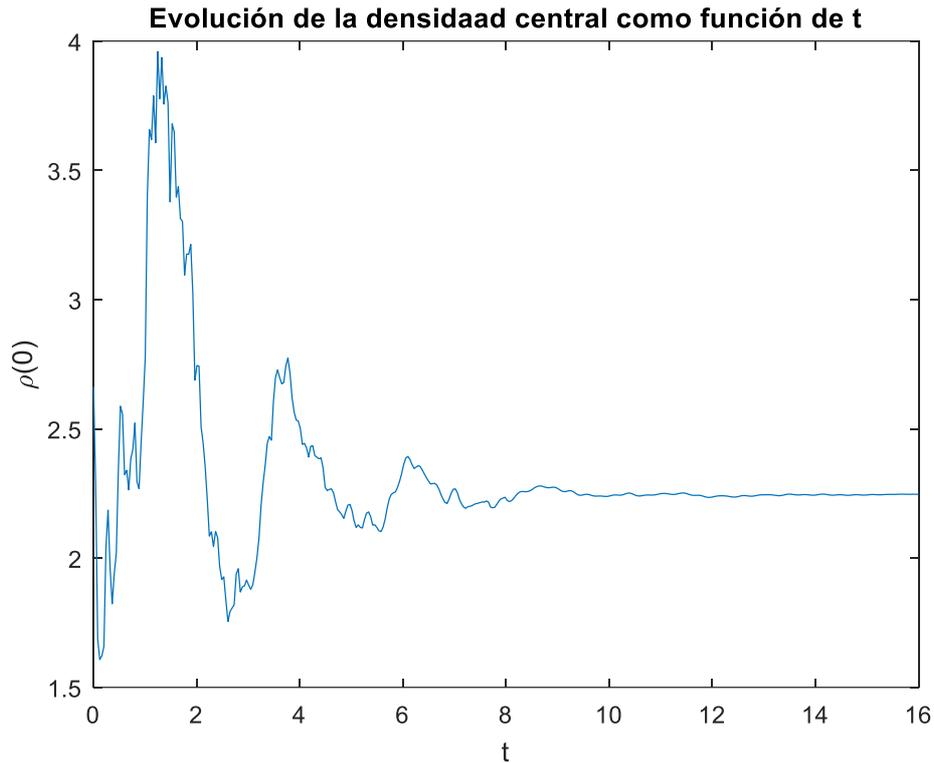


Figura (5-19)

5.3 Códigos Matlab

Para acabar, se presentan los códigos empleados para la simulación.

En primer lugar, se presenta el programa utilizado para la simular como alcanza el equilibrio una estrella a partir de una configuración aleatoria de radio R .

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----Programa principal-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Modelo simplificado de Estrella 2D
clear all
close all
rng('default')

%% Datos

N   = 300;           %Numero de particulas
d   = 2;            %Dimension del problema
h   = 0.06/sqrt(N/1000); %Smoothing length
dt  = 0.04;         %Time step
nu  = 3;            %Coeficiente de viscosidad
k   = 0.1;          %Constante de presion
n   = 1;            %Indice politropico
M   = 2;            %Masa de la estrella
m   = M/N;          %Masa de cada partícula
R   = 0.75;         %Radio de la estrella

max_time_step = 400;

```

```

%Termino gravitacional

lambda = 2*k*(pi^(-1/n)) * ((M*(1+n)/(R^2))^(1+1/n))/M;

%% Creamos puntos random dentro de un circulo de radio R

x = zeros(1,d); % Primer punto en el centro
ii = 2;         % Iniciamos el indice

while length(x(:,1))<N

    y=-R+(R+R).*rand(1,d);

    if norm(y)^2<=R^2

        x(ii,1) = y(1);
        x(ii,2) = y(2);
        ii      = ii+1;

    end

end

%% Bucle
%Inicializamos las variables

v_mhalf = zeros(N,d);
a        = zeros(N,d);
rho0     = zeros(N,1); %vector para almacenar el maximo valor de la densidad
time     = 0;

for i=1:max_time_step

    disp(i)

    %Actualiacion posicion mediante el algoritmo Leap frog
    v_phalf = v_mhalf+a*dt;
    x       = x+v_phalf*dt;
    v       = 0.5*(v_mhalf+v_phalf);
    v_mhalf = v_phalf;

    %Actualizamos las variables
    rho      = calculate_density(x,m,h,d,N);
    P       = k*rho.^(1+1/n); %relacion de fluido politropico
    a       = calculate_acceleration(x,v,m,rho,P,nu,lambda,h,d,N);
    rho0(i) = max(rho);

    %Animación
    %2D
    figure(1)
    scatter(x(:,1),x(:,2),[],rho,'filled')
    axis([-1 1 -1 1 ])
    hold on
    colorbar
    plot(xc,yc,'r')
    pause(0.001)
    hold off

end

```

```

%% Evolución de rho_max

figure(2)
hold on
plot(linspace(0,16,400),rho0,'r')
xlabel('t')
ylabel('\rho(0)')
title('Evolución de la densidad central como función de t')
%legend('\nu=1','\nu=3')

%% Representar rho(r) en el equilibrio mediante un promediado

Nr      = 1000;
r       = zeros(Nr,d);
r(:,1) = linspace(0,R,Nr)';
rho_r   = zeros(Nr,1);

for i=1:Nr

    sum=0;

    for j=1:N,

        uij = r(i,:)-x(j,:);
        sum = sum+m*spline_kernel(uij,h,d);

    end

    rho_r(i,1)=sum;

end

%Curva analitica

d_ana = (lambda/(2*k*(1+n))*(R^2-r.^2)).^n;

%Representación de ambas curvas

figure(3)
hold on
plot(r(2:end,1),rho_r(2:end),'m')

hold on
plot(r(2:end,1),d_ana(2:end,1),'--r')
xlabel('r')
ylabel('\rho(r)')
title('Perfil de densidad en el equilibrio')
legend('h=0.04/sqrt(N/1000)','Sol analítica','h=0.03/sqrt(N/1000)')

%% Comparar con valor analitico en rho(0)

rho_maxana = lambda/(4*k)*R^2; %Densidad en el centro de la estrella, valido
para n politropico igual a 1
disp([rho_maxana max(rho_r)])

```

Muy parecido a este programa es el utilizado para la colision entre estrellas, únicamente variando las condiciones iniciales.

```
% Modelo de Estrella simplificado, collision entre estrellas
```

```
clear all
close all
rng('default')
```

```
%% Datos
```

```
N = 500; %Numero de particulas
d = 2; %Dimension del problema
h = 0.04/sqrt(N/1000); %Smoothing length
dt = 0.04; %Time step
nu = 1; %Coeficiente de viscosidad
k = 0.1; %Constante de presion
n = 1; %Indice politropico
M = 2; %Masa de la estrella
m = M/N; %Masa de cada partícula
R = 0.75; %Radio de la estrella
```

```
max_time_step = 400;
```

```
%Termino gravitacional
```

```
lambda = 2*k*(pi^(-1/n)) * ((M*(1+n)/(R^2))^(1+1/n))/M;
```

```
%% Creamos puntos primera estrella
```

```
x = zeros(1,d); % Primer punto en el centro
ii = 2; % Iniciamos el indice
v_mhalf = zeros(1,d);
```

```
while length(x(:,1))<N*3.5/5
```

```
    y=-R+(R+R).*rand(1,d);
    v_mhalf(ii,d) = 0.2; % Velocidad dirección estrella 2
```

```
    if norm(y)^2<=R^2
```

```
        x(ii,1) = y(1);
        x(ii,2) = y(2);
        ii = ii+1;
```

```
    end
```

```
end
```

```
%Particulas Estrella 2, con offset
```

```
while length(x(:,1))<N
```

```
    y=1.65/2+(3.15-1.65)/2.*rand(1,d);
```

```
    x(ii,1) = y(1);
    x(ii,2) = y(2);
    v_mhalf(ii,d) = -0.2; % Velocidad dirección estrella 1
    ii = ii+1;
```

```

    if norm(y)^2<=R^2

        x(ii,1) = y(1);
        x(ii,2) = y(2);
        ii      = ii+1;

    end

end

%% Bucle

%Inicializamos las variables

a      = zeros(N,d);
rho0   = zeros(400,1); %vector para almacenar el maximo valor de la densidad
p0     = zeros(400,1); %vector para almacenar el maximo valor de la presión

%Recorremos en el tiempo

for i=1:max_time_step

    disp(i)

    %Actualiacion posicion con el algoritmo Leap frog

    v_phalf = v_mhalf+a*dt;
    x        = x+v_phalf*dt;
    v        = 0.5*(v_mhalf+v_phalf);
    v_mhalf = v_phalf;

    %Actualizamos las variables

    rho      = calculate_density(x,m,h,d,N);
    P        = k*rho.^(1+1/n); %relacion de fluido politropico
    a        = calculate_acceleration(x,v,m,rho,P,nu,lambda,h,d,N);
    rho0(i)  = max(rho);

    p0(i)    = max(P);

    %Animación
    %2D

    figure(1)
    scatter(x(1:3.5*N/5,1),x(1:3.5*N/5,2),'filled')
    hold on
    scatter(x(3.5*N/5+1:end,1),x(3.5*N/5+1:end,2),'filled')
    axis([-2 2 -2 2])
    pause(0.001)
    hold off

end

```

```

%% Evolución de rho_max

figure(2)
plot(linspace(0,16,400),rho0)
xlabel('t')
ylabel('\rho(0)')
title('Evolución de la densidad central como función de t')
%% Evolucion de P_max

figure(7)
plot(linspace(0,16,400),p0)
xlabel('t')
ylabel('P(0)')
title('Evolución de la presión central como función de t')

```

Dentro de ambos programas se emplean diversas funciones que se detallan a continuación. Entre ellas están la función que calcula el nuevo valor de densidad y el de aceleración a través de la ecuación del movimiento. Además, dentro de estas funciones se emplean las funciones kernel, tanto Gaussian Kernel como Spline kernel (ya presentado en el capítulo del tubo de choque).

```

function rho=calculate_density(x,m,h,d,N)

%CALCULATE_DENSITY- function to calculate density
%                   Format:rho=calculate_density(x,m,h,d,N)
%                   x-particle positions
%                   m-mass of each particle
%                   h-smoothing length

rho = zeros(N,1);

for i=1:N

    rho(i) = m*spline_kernel(0,h,d); %initialize with i=j contribution

    for j=i+1:N

        uij      = x(i,:)-x(j,:);
        rho_ij   = m*spline_kernel(uij,h,d);
        %Add contribution to density
        rho(i) = rho(i)+rho_ij;
        rho(j) = rho(j)+rho_ij;

    end

end

end

end

```

```

function [a]=calculate_acceleration(x,v,m,rho,P,nu,lambda,h,d,N)

%CALCULATE_ACCELERATION- function to calculate acceleration due to
%                          pressure, damping and gravity.
%
Format:a=calculate_acceleration(x,v,m,rho,P,nu,lambda,h,d,N)
%                          x-particle positions
%                          m-mass of each particle
%                          h-smoothing length
a      = zeros(N,d);

%Damping and gravity

for i=1:N

    a(i,:) = a(i, :)-nu*v(i, :)-lambda*x(i, :);

end

%Pressure

for i=1:N

    for j=i+1:N

        uij  = x(i, :)-x(j, :);

        p_a  = -m*(P(i)/(rho(i)^2)+P(j)/(rho(j)^2))*spline_kernelp(uij,h,d);

        a(i,:) = a(i, :)+p_a;
        a(j,:) = a(j, :)-p_a;

    end

end

end

function w=gaussian_kernel(r,h,d)

%GAUSSIAN_KERNEL- Smoothing kernel function, best option for reproducing
%                  physical answers.
%                  Format:w=gaussian_kernel(r,h,d)
%                  d-dimension of the problem
%                  h-kernel length
w=((1/(h*sqrt(pi)))^d)*exp(-norm(r)^2/h^2);
end

function w=gaussian_kernelp(r,h,d)

%GAUSSIAN_KERNELP- Gradient of Smoothing kernel function, best option for
reproducing
%                  physical answers.
%                  Format:w=gaussian_kernelp(r,h,d)
%                  d-dimension of the problem
%                  h-kernel length

w=(1/(h*sqrt(pi)))^d*exp(-norm(r)^2/h^2)*(-2*(r)/h^2);

end

```

6 CONCLUSIONES Y DESARROLLOS FUTUROS

“No basta tener buen ingenio, lo principal es aplicarlo bien”

-René Descartes-

A lo largo de la memoria se han expuesto las bases teóricas del método Smoothed Particle Hydrodynamics (SPH), justificando su uso, su implementación en Matlab y los resultados que ofrece en diversas aplicaciones sencillas como en el tubo de choque o el modelo simplificado de estrella. Tras ello, se concluye que el método SPH es un método robusto, relativamente sencillo de implementar, el cual ofrece una visión intuitiva de la física que gobierna el problema.

La precisión del método SPH tanto para la aplicación del tubo de choque como para el modelo simplificado de estrellas ha permitido obtener resultados que concuerdan con la bibliografía consultada. Además, en un tiempo que no resulta excesivamente elevado.

Se ha tratado del primer Trabajo Fin de Grado realizado en la Escuela Superior Técnica de Ingeniería de Sevilla, sobre el SPH, donde se ha pretendido ilustrar las bases del método y servir como primer paso para futuros trabajos donde se extiendan las aplicaciones a desarrollar.

De este modo, las líneas futuras de investigación que se proponen son:

- El estudio de problemas en los que se incluyan condiciones de contorno. Puede ser el caso de fluidos confinados en cavidades, donde habría que hacer uso de partículas virtuales, las cuales realizarían la función de pared.
- Estudio de la variación del smoothing length en cada iteración, pudiendo ser distinto en cada partícula. Llevando a resultados aún más precisos.
- Simulación de problemas donde se estudie la turbulencia y convecciones.

REFERENCIAS

- [1] G.R.Liu & M.B.Liu : Smoothed Particle Hydrodynamics a meshfree particle method, 2003
- [2] Philip Mocz: Smoothed Particle Hydrodynamics: Theory, Implementation, and application to Toy Stars 2011.
- [3] M.M. Woolfson: The Origin and Evolution of the Solar System, 2000.
- [4] J.J. Monaghan & R.A. Gingold : Shock Simulation by the Particle Method, Journal of Computational Physics ,1983.
- [5] G.R.Liu & M.B.Liu : Smoothed Particle Hydrodynamics (SPH) :an Overview and Recent Developments, 2010
- [6] J.J. Monaghan & R.A. Gingold : Smoothed Particle Hydrodynamics: theory and application to non-spherical stars, Mon.Not.R.Soc ,1977
- [7] J.J. Monaghan & J.C Lattanzio : A refined particle method for astrophysical problems, Astron.Astrophys, 1985
- [8] J.J. Monaghan: Smoothed Particle Hydrodynamics, Annu.Rev.Astron.Astrophys, 1992.
- [9] J.J. Monaghan : Smoothed Particle Hydrodynamics, Institute of Physics Publishing, 2005.