



SOFTWARE

FlowMove

Autores

Diego Valencia-Enríquez¹
Javier Revelo-Fuelagán²

¹Centro de Investigación y Desarrollo Tecnológico en Ciencias Aplicadas (CIDTCA)

²Departamento de Electrónica, Facultad de Ingeniería, Universidad de Nariño, Pasto,
Colombia

Lista de Contenidos

1	Introducción	3
1.1	Sobre el Método SPH	4
1.2	Ventajas y desventajas del método SPH	4
1.3	Aplicaciones	4
2	Método SPH	8
2.1	Aproximación al dominio continuo	9
2.2	Formulación SPH para las ecuaciones de Navier-Stokes	10
2.3	Ecuaciones extendidas	12
2.3.1	Conservación de Masa:	12
2.3.2	Conservación de Momentum	12
2.4	Conservación de momento expandida con el tensor de tasa de deformación de corte	13
2.4.1	Ecuaciones de la tasa de deformación por corte $\varepsilon^{\alpha\beta}$	14
2.4.2	Tasa de deformación viscosa desglosada	14
3	Funciones de Suavizado	17
3.1	Núcleo Gaussiano (Gaussian kerne)	17
3.2	Núcleo de Spline Cúbico (Cubic Spline Kernel:)	17
3.3	Núcleo de Spline Quintic (Quintic Spline Kernel):	17
4	Búsqueda de partículas vecinas	18
4.1	Método Grid Hashing	18
5	Método de salto de rana	20
6	Modelo reológico	21
6.1	Modelo de Herschel-Bulkley	21
6.2	Modelos Reológicos (Herschel-Bulkley y Bingham)	22
6.2.1	Formulación tensorial - SPH del modelo de Herschel-Bulkley	24
7	Descripción de Rutinas	27
7.1	Rutina <code>sph</code>	27
7.2	Rutina <code>input</code>	30
7.3	Rutina <code>time_integration</code>	31
7.4	Rutina <code>fuentes</code>	32
7.5	Rutina <code>single_step</code>	34
7.6	Rutinas <code>neighboring_search</code> y <code>neighborin.Grid.Hashing</code>	36
7.7	Rutinas <code>wijdwij</code> y <code>kernel</code>	37
7.8	Rutina <code>density</code>	37
7.9	Rutina <code>presioni</code>	38
7.10	Rutina <code>momento</code>	38

7.11 Rutina <code>sph_presion</code>	40
7.12 Rutina <code>tasa_deformacion_epsilon</code>	42
7.13 Rutina <code>tasa_deformacion_epsilon2</code>	43
7.14 Rutina <code>viscous_force</code>	43
7.15 Rutina <code>viscous_force_mud</code>	44
7.16 Rutina <code>viscous_model</code>	45
7.17 Rutina <code>external_force</code>	45
7.18 Archivo de Parámetros del Código SPH, <code>param.inc.</code>	45

1 Introducción

Los movimientos de masa, como los flujos de lodos o lahares, constituyen algunos de los fenómenos naturales más catastróficos que captan la atención de ingenieros, investigadores y organismos gubernamentales. Estudios indican que estos movimientos afectan áreas con una planificación territorial deficiente y su principal desencadenante son las precipitaciones (Enríquez 2012).

Un ejemplo reciente tuvo lugar en el municipio de Rosas, Cauca, donde el aumento de las lluvias y la falla geológica conocida como falla Mosquerillo-La Tetilla del Sistema de Fallas de Romeral resultaron en un deslizamiento de tierra que destruyó al menos 150 viviendas, terrenos agrícolas, áreas boscosas, infraestructura de acueductos y eléctrica, además de dañar un tramo de 300 metros de la vía Panamericana. Esta catástrofe dejó incomunicada la región sur del país y causó pérdidas económicas significativas (Carrillo et al. 2023).

Otro ejemplo relevante es el desastre natural en Mocoa, que ocurrió en marzo de 2017. Este evento, conocido como avenida torrencial, implica el aumento descontrolado de los niveles de agua en una cuenca o microcuenca. Los informes indican alrededor de 300 muertes, 400 heridos, 200 personas desaparecidas y alrededor de 1,000 personas afectadas (Santamaría et al. 2018).

Estos eventos de movimientos de masa también se presentan a nivel mundial. Por ejemplo, en Nicaragua en 1998, un lahar en el Volcán Casita resultó en unas 2,500 muertes y la devastación de las localidades de El Porvenir y Rolando Rodríguez (Scott et al. 2005). En 2005, debido a las lluvias asociadas al Huracán Stan, el Río Coatán en México se desbordó, generando un flujo de lodo que destruyó alrededor de 2,000 viviendas, afectó a 100,000 personas y causó daños en la red ferroviaria y cuatro puentes (Murcia & Macías 2009).

Por otro lado, en el trabajo realizado por Chaparro-Cordón et al. (2020), se recopiló y analizó información sobre deslizamientos en las regiones de Mocoa, Cajamarca, Villavicencio y Popayán, conocidos como Inventarios de Deslizamientos, con el apoyo del Servicio Geológico Colombiano (SGC) entre 2015 y 2018. Se encontró que la mayoría de los deslizamientos ocurren en pendientes empinadas que van desde 35° a 55°, y su distribución espacial está estrechamente relacionada con la actividad tectónica y la geomorfología de la región.

Por lo tanto, el estudio y modelado de la dinámica de flujos de lodos puede servir como base para la planificación de la gestión del riesgo por parte de las autoridades. Los modelos numéricos son capaces de simular fenómenos específicos y predecir el movimiento de masa de los flujos, estimar volúmenes de deslizamientos depositados en una corriente, espesores de flujo a distancias y tiempos determinados, así como proporcionar información sobre velocidades, presiones y otros parámetros físicos (Enríquez 2012). En esta línea, existen códigos comerciales y de acceso libre utilizados por el SGC, como RiverFlow2d©, FLO2D© y Titan2d.

Por otro lado, un flujo de lodos es un flujo esencialmente bifásico: la fase líquida está conformada por agua y la fase sólida son los depósitos de suelos de deslizamientos de

laderas o de erupciones volcánicas, la cual es difícil implementar en los anteriores códigos mencionados. Además, los Software anteriormente mencionados resuelven las ecuaciones mediante métodos de diferencias finitas estableciendo la correspondiente mallas en modelos en dos dimensiones lo que genera limitaciones en los resultados obtenidos. Por lo que en este estudio se propone implementar el método de SPH (Smoothed Particle Hydrodynamics, por sus siglas en inglés), para modelar este tipo de fluidos multifase en tres dimensiones para tener una mejor aproximación de las características físicas del fluido y del área de cobertura del movimiento de masa.

1.1 Sobre el Método SPH

El método SPH (por sus siglas en inglés, *Smoothed Particle Hydrodynamics*) es un método Lagrangiano, fue desarrollado inicialmente para solucionar problemas de astrofísica (Gingold & Monaghan 1977; Lucy 1977) y en los últimos años ha sido aplicado en problemas de dinámica de fluidos (Ye et al. 2019).

El método SPH discretiza el fluido como un conjunto de puntos o nodos llamados partículas, las ecuaciones de movimiento son resueltas usando el formalismo Lagrangiano, calculando el espacio de fase (velocidad, posición, densidad y presión) para cada partícula como la interpolación de los valores de las partículas vecinas más cercanas usando las ecuaciones diferenciales acopladas de la hidrodinámica. La interpolación se realiza estableciendo una función de suavizado, conocida como función núcleo o *kernel*, esta es usada para ejecutar la transición de un dominio continuo a uno discreto; el dominio de esta función es definido por la longitud de suavizado que indica la distancia de interacción entre las partículas.

1.2 Ventajas y desventajas del método SPH

Una de las ventajas principales en el método SPH está dada por la naturaleza del Lagrangiano ya que permite realizar simulaciones de flujos muy complejos; el cálculo computacional es más eficiente ya que las partículas son las que representan el fluido y las condiciones de frontera; a diferencia del método de elementos finitos, en el método SPH no hay restricción para plantear la geometría del problema y es relativamente más fácil incluir procesos físicos adicionales en las ecuaciones de movimiento (Ye et al. 2019).

Por otro lado, una de las desventajas que se puede notar en el método SPH es que la implementación de las condiciones de frontera es una tarea difícil, de hecho es un tópico de estudio en la comunidad SPH; sin embargo, con ayuda de las herramientas topográficas, las condiciones de frontera por donde se puede mover el flujo facilitarán esta tarea.

1.3 Aplicaciones

En el comienzo, el método SPH fue desarrollado para problemas astrofísicos tales como la rotación de estrellas (Lucy 1977), la componente de gas en galaxias (Springel 2005), entre otros. Así mismo, el método ha sido extendido a un amplio rango de problemas

en mecánica de fluidos y sólidos debido a la facilidad que tiene de incorporar diferentes procesos físicos dentro de la formulación SPH.

Los fluidos biológicos son de interés en el estudio de procesos naturales, es así como los primeros estudios sobre el movimiento de la sangre en los vasos sanguíneos fueron experimentales (Schmid-Schönbein et al. 1980; Pries et al. 1990), los subsecuentes se fueron desarrollando con el uso de simulaciones numéricas, en algunos casos utilizando el método SPH (Polwaththe-Gallage et al. 2016a,b).

Un fluido no newtoniano es un fluido que no sigue la ley de viscosidad de Newton. La viscosidad de los fluidos no Newtonianos depende de la velocidad de corte o del historial de velocidad de corte. Por tanto, los fluidos no Newtonianos también pertenecen a la categoría de fluido complejo. La primera simulación de un fluido no Newtoniano fue desarrollada por Ellero et al. (2002) y subsecuentes estudios en el tema se han desarrollado en los años siguientes (Fan et al. 2010; Zhang et al. 2020), etc. Con estos desarrollos, el método SPH se ha aplicado en los estudios sobre flujo de lodo o flujo de escombros (Laigle et al. 2007; Wang et al. 2016), partículas sólidas suspendidas en el flujo cortante de fluido Oldroyd-B (Hashemi et al. 2011), fusión de polímero (Xu 2013), Ley de potencias de fluido a través de medios porosos (Vakilha & Manzari 2008), y en el proceso de impresión 3D (Bertevas et al. 2018), entre otros trabajos.

La implementación de superficies libres en el método SPH es más simple que en métodos basados en mallas (Ye et al. 2019). Hay tres casos representativos, que son chapoteos (*sloshing*), ondas estacionarias y propagación de ondas. Chapoteos consiste en el movimiento violento de fluidos dentro de un tanque (Shao et al. 2012). Ondas estacionarias es un típico problema de fluido de superficie libre, donde la energía total de las partículas del fluido se hace cero y la onda estacionaria es amortiguada a medida que transcurre el tiempo (Antuono et al. 2011). La propagación y la ruptura de las olas son de gran importancia en la ingeniería costera y oceánica, Zhang et al. (2018) usa el método SPH para estudiar la oscilación de las olas. También, se estudia el comportamiento de “grandes” olas, en aguas poco profundas (Del Guzzo & Panizzo 2007; Vacondio et al. 2010; De Leffe et al. 2010), o debido a deslizamientos de tierra submarinos (Panizzo & Dalrymple 2005; Panizzo et al. 2007; Rogers & Dalrymple 2008; Capone et al. 2010). En el transporte y suspensión de sedimentos bajo el efecto de las olas utilizando la forma lagrangiana de la ecuación de difusión (Zou & Dalrymple 2007), o en la suspensión de sedimentos en tanques industriales (Fourtakas et al. 2013). En simulación SPH para el lavado de sedimentos inducido por flujos de agua rápida (Manenti et al. 2012). En el modelado de inundaciones (Vacondio et al. 2016; Viero & Valipour 2017).

La interacción de fluidos y estructuras sólidas se puede realizar fundamentalmente por métodos basados en mallas o métodos SPH. Los métodos basados en mallas, los cuales son usados por los códigos mencionados en las secciones anteriores, son complicados de implementar, en tanto que el método SPH es adecuado para ese tipo de simulaciones ya que es de naturaleza lagrangiana (Ye et al. 2019). Por ejemplo, bin Liu et al. (2013) estudia la interacción de fluidos con objetos rígidos moviéndose, Zhang & Liu (2017) implementa el método SPH para modelar problemas de impacto de alta velocidad en espacios bidimensionales y tridimensionales.

En la interacción entre olas y estructuras costeras Gomez-Gesteira & Dalrymple (2004) usa el método SPH 3D para el impacto de las olas en estructuras altas, en simulación de movimiento acoplado de onda progresiva y un muro flotante (Shao & Gotoh 2004), en análisis de los fenómenos de desbordamiento en cubiertas horizontales (Gómez-Gesteira et al. 2005), en el estudio de la turbulencia en zonas de olas rompiendo (Dalrymple & Rogers 2006), en barreras de protección para mitigar la fuerza y el impulso ejercido por grandes olas (Crespo et al. 2007; Altomare et al. 2014).

El método SPH es eficaz en rastrear la interfaz en simulaciones de flujo multifase, el cual es bastante difícil en métodos basados en mallas. Por ejemplo, simulación de dos fases; en aire y agua e interacción de estructuras con onda (Cuomo et al. 2007), para flujos multifásicos rápidos y propagación de ondas de choque (Manenti et al. 2011), para modelado multifásico de hidrodinámica violenta utilizando (SPH) en unidades de procesamiento de gráficos (gpus) (Mokos 2014), para evaporar el flujo multifásico (Yang & Kong 2017) donde estudia el calentamiento y la transferencia de masa a través de la interfaz líquido y gas, para modelar la interacción de múltiples fluidos con la superficie libre y su aplicación en derrames de petróleo (Duan et al. 2017), para modelado de flujos multifase en desmantelamiento nuclear utilizando SPH (Fourtakas 2014), entre otros.

Los eventos más comunes en los procesos eruptivos de los volcanes son los fluidos de lodo, magma y los fluidos piroclásticos, que pueden ser devastadores si una población se asienta en los alrededores de un volcán (Enríquez 2012). Por ejemplo, el Volcán Galeras, que está ubicado en Nariño - Colombia, tiene en los alrededores una gran cantidad de población lo que conlleva a representar un riesgo debido a los fluidos que este puede generar. Gallo et al. (2011) modela flujos Piroclásticos en el Volcán Galeras con el Código TiTAN2D, el código establece el área que puede ser afectada debido a una columna volcánica. El programa TITAN2D se basa en un modelo de profundidad promedio para medio continuo incompresible, en el cual se cumplen las leyes de Coulomb para un flujo granular de aguas poco profundas, las ecuaciones que gobiernan el movimiento del fluido son resueltas mediante métodos de malla, sin embargo no es susceptible cuando se presentan obstáculos en el movimiento en el fluido, la cual una implementación con SPH puede resolver este problema. Cáceres (2011) presenta tres mapas de riesgo en los alrededores del Volcán Galeras mediante modelos geofísicos donde se involucra la población, la economía y la infraestructura. Finalmente, Enríquez (2012) analiza las condiciones de vulnerabilidad ante flujos de lodo. En este sentido, se encuentran muy pocos estudios de la simulación de este tipo de fluidos en la región de Nariño.

Por otro lado, algunos estudios sobre flujos de lodo y piroclásticos se citan a continuación. Haddad et al. (2016) realiza un modelado hidrodinámico de partículas suavizadas de flujos de escombros volcánicos aplicado a los lahares del desfiladero de Huiloac en el volcán de Popocatepetl, México. Cao et al. (2018) proporciona la primera simulación de penachos volcánicos basada en partículas SPH para pronosticar el movimiento de cenizas volcánicas.

Adicionalmente, en los últimos años se ha optado por resolver los problemas ingenieriles mediante simulaciones computacionales ya que el diseño de experimentos para la solución de estos problemas conlleva costos altos, por ejemplo para entender el colapso de una

columna de agua sobre un tanque (Lee et al. 2010), para entender la dinámica en un salto hidráulico de fluidos (López et al. 2010), o para entender el comportamiento de un fluido en turbinas Pelton (Marongiu et al. 2010).

2 Método Smoothed Particle Hydrodynamics (SPH)

La esencia de este método Lagrangiano radica en discretizar el dominio continuo en un número finito de partículas que obtienen las magnitudes físicas de interés mediante interpolaciones ponderadas de las magnitudes en las partículas adyacentes. Solamente aquellas partículas que se encuentren dentro del dominio de influencia (a una distancia máxima de κh de la partícula fija considerada) contribuirán al comportamiento de las magnitudes físicas en la partícula fija. La Figura 1 ilustra una representación gráfica de las partículas Lagrangianas dentro del dominio de influencia (circunferencia con un radio igual a κh). SPH se basa en la identidad matemática, válida para una función definida y continua $f(X)$, como se muestra en la siguiente ecuación:

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

donde $\delta(x - x')$ es la función Delta de Dirac definida por

$$\delta(x - x') = \begin{cases} \infty, & \text{si } x = x' \\ 0, & \text{si } x \neq x' \end{cases} \quad (2)$$

$x = (x, y, z)$ es la posición del punto fijo, $x' = (x', y', z')$ es la posición del punto variable, $f(x)$ es el valor de la función escalar en el punto fijo, $f(x')$ es el valor de la función escalar en el punto variable, Ω es el dominio y dx' es el elemento de volumen infinitesimal.

La función $\delta(x - x')$ no se puede usar en modelos numéricos debido a que aplica en un solo punto, por lo que se reemplaza por una función de suavizado, “*smoothing function*”, $W(x - x', h)$, con un espacio finito de dimensión h , por lo que la aproximación de la función de núcleo, que se denominaría *kernel*, se vuelve:

$$\langle f(x) \rangle = \int_{\Omega} f(x') W(x - x', h) dx', \quad (3)$$

donde $\langle f(x) \rangle$ es el valor aproximado de la función escalar f en el punto x con una longitud de suavizado h y $W(x - x', h)$ es el *kernel* evaluado en $(x - x')$.

En el método SPH, como se muestra en Liu & Liu (2010), el *kernel* da el valor de la propiedad física f , debido a la interpolación con las demás partículas vecinas, en este sentido la mayor contribución al *kernel* debe ser de las partículas más cercanas, por lo que debe satisfacer las siguientes propiedades:

- Debe ser positivo: $W(x - x', h) \geq 0$.
- Debe ser simétrico: $W(x - x', h) = W(x' - x)$.
- Debe ser normalizado en el dominio de influencia $\int_{\Omega} W(x - x', h) dx' = 1$.
- Debe tener soporte compacto, es decir la función de suavizado tiene un rango de influencia limitado en el espacio, más allá del cual su valor disminuye rápidamente. En

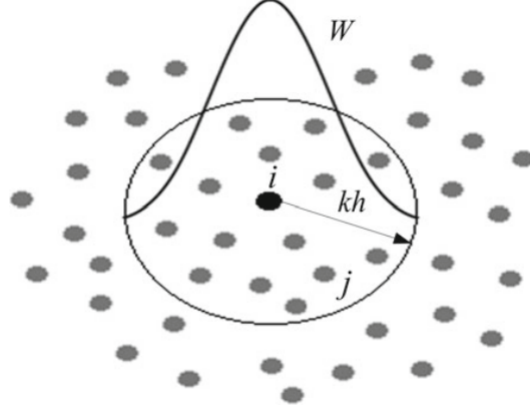


Figure 1: Representación gráfica del kernel y su dominio de influencia. La partícula de referencia 'i' (círculo central negro) tiene sus propiedades físicas influenciadas por las partículas vecinas 'j', que están dentro del dominio de influencia. Tomado de Fraga Filho et al. (2019)

otras palabras, las partículas cercanas dentro de este intervalo tendrán una influencia significativa en la aproximación de una propiedad física en un punto en particular, mientras que las partículas fuera de este intervalo tendrán una influencia despreciable o cero. Por lo tanto, $W(x - x', h) = 0$ cuando $|x - x'| > kh$, donde kh es el radio del soporte de dominio.

- Convergencia de la función Delta de Dirac: $\lim_{h \rightarrow 0} W(x - x', h) = \delta(x - x')$. Significa que a medida que los puntos $x - x'$ se acerquen el 'kernel' se acerca a la función Delta de Dirac.
- Decaimiento: Su valor disminuye monótonamente a medida que el punto evaluado se aleja del punto central del dominio de influencia.
- Suavidad: La función de suavizado debe ser suficientemente suave.

2.1 Aproximación al dominio continuo

El método SPH discretiza el problema al representarlo como un sistema de partículas, lo que permite abordar el problema dentro de un dominio computacional que consta de un número finito de partículas. En este contexto, no es factible emplear la aproximación del kernel en una forma continua, tal como se presenta en la ecuación 3. Por lo tanto, es necesario aplicar una discretización de esta ecuación. De acuerdo con (Liu & Liu 2010), esta discretización se expresa de la siguiente manera:

$$\langle f_i \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f_j W(x_i - x_j, h), \quad (4)$$

donde f_i es la aproximación del valor de la función en la posición de la partícula fija, f_j es el valor aproximado de la función en la posición de la partícula vecina, m_j es la masa de la partícula vecina, ρ_j es la densidad de la partícula vecina, x_i es la posición de la partícula fija, x_j es la posición de la partícula vecina, N es el número de partículas y $W(x_i - x_j, h)$ es el kernel evaluado en la posición $(x_i - x_j, h)$. En ese caso se ha cambiado a $W(x - x', h)$ por $W(x_i - x_j, h)$.

Además, se puede obtener la discretización del gradiente y el Laplaciano de la función escalar (Fraga Filho et al. 2019), el cual es

$$\nabla f_i = -\rho_i \sum_{j=1}^N m_j \left(\frac{f_i}{\rho_i^2} + \frac{f_j}{\rho_j^2} \right) \nabla W(x_i - x_j, h), \quad (5)$$

y

$$\nabla^2 f_i = 2 \sum_{j=1}^N \frac{m_j}{\rho_j} (f_i - f_j) \frac{\partial W(\vec{r}_i - \vec{r}_j, h)}{\partial r} \frac{1}{|\vec{r}_{ij}|}, \quad (6)$$

donde r es la dirección radial, $W(\vec{r}_i - \vec{r}_j, h)$ es la función de suavizado en 3D y $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$.

2.2 Formulación SPH para las ecuaciones de Navier-Stokes

Usando las aproximaciones anteriores, se obtiene la formulación SPH para las ecuaciones de Navier-Stokes (N-S). Entonces, las ecuaciones que controlan la dinámica de un fluido en general están dadas por:

$$\frac{D\rho}{Dt} = -\rho \frac{\partial \mathbf{v}^\beta}{\partial \mathbf{x}^\beta}, \quad (7)$$

$$\frac{D\mathbf{v}^\alpha}{Dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial \mathbf{x}^\beta} + F, \quad (8)$$

$$\frac{De}{Dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{\partial \mathbf{v}^\alpha}{\partial \mathbf{x}^\beta}, \quad (9)$$

donde las variables en negrilla representan vectores de esas cantidades físicas, los superíndices en alfabeto Griego α y β son usados para representar la dirección de las coordenadas, las cual son coordenadas independientes, la suma en las ecuaciones se realiza sobre índices repetidos, y las derivadas totales con respecto al tiempo se toman en el marco lagrangiano móvil, e la energía interna, \mathbf{v}^α la componente de velocidad y $\sigma^{\alpha\beta}$ es el tensor de esfuerzos total. F representa las fuerzas externas, como la gravedad (g^α) y los contornos o paredes que limitan el movimiento del fluido.

El tensor de esfuerzo total es la suma de la presión isotrópica p y el esfuerzo por viscosidad τ , de forma que el tensor de esfuerzo total se escribe como

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta}, \quad (10)$$

donde

$$\tau^{\alpha\beta} = \mu \varepsilon^{\alpha\beta}, \quad (11)$$

y ε es la tasa de deformación por corte local y μ es la viscosidad dinámica, para el agua a 20°C toma un valor de 10^{-3} Pa-s. Para fluidos Newtonianos, la tasa de deformación por corte está dada por

$$\varepsilon^{\alpha\beta} = \frac{\partial \mathbf{v}^\beta}{\partial \mathbf{x}^\alpha} + \frac{\partial \mathbf{v}^\alpha}{\partial \mathbf{x}^\beta} - \frac{2}{3} (\nabla \cdot \mathbf{v}) \delta^{\alpha\beta}. \quad (12)$$

El siguiente paso ahora, es reescribir las ecuaciones N-S, y las ecuaciones 10 y 12 con la formulación SPH usando el proceso descrito en la sección 2.1. Se puede mostrar que las ecuaciones N-S se pueden reescribir como

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j \mathbf{v}_{ij}^\beta \cdot \left(\frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} \right), \quad (13)$$

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} + g^\alpha, \quad (14)$$

$$\frac{De_i}{Dt} = \frac{1}{2} \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \mathbf{v}_{ij}^\beta \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} + \frac{\mu_i}{2\rho_i} \varepsilon_i^{\alpha\beta} \varepsilon_i^{\alpha\beta}, \quad (15)$$

donde $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. Así mismo, a aproximación SPH para la densidad está dada por

$$\rho_i = \sum_{j=1}^N m_j W_{ij}. \quad (16)$$

Sustituyendo la ecuación (10) y (11) en la ecuación (15), la derivada total se puede escribir como

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\alpha} + \sum_{j=1}^N m_j \left(\frac{\tau_i^{\alpha\beta}}{\rho_i^2} + \frac{\tau_j^{\alpha\beta}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} + g^\alpha, \quad (17)$$

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\alpha} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{\alpha\beta}}{\rho_i^2} + \frac{\mu \varepsilon_j^{\alpha\beta}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} + g^\alpha, \quad (18)$$

El primer término del lado derecho de esta ecuación representa la aproximación SPH para la presión, mientras que el segundo representa la aceleración debido a la fuerza de viscosidad. La presión se calcula mediante la ecuación de estado para la presión hidrostática, que se presenta por

$$p = B \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right], \quad (19)$$

donde, la constante $B = c_0^2 \rho_0 / \gamma$ es la densidad de referencia, $c_0 = \beta \sqrt{2gh_{max}}$ es la velocidad del sonido para la densidad de referencia, $\beta = 5$ es una constante arbitraria y $\gamma = 7$ un parámetro sin dimensiones.

2.3 Ecuaciones extendidas

A continuación se desglosa las ecuaciones de Navier-Stokes con formulación SPH en sus componentes para un sistema cartesiano en x , y , z .

2.3.1 Conservación de Masa:

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j \left(\mathbf{v}_{ij}^x \frac{\partial W_{ij}}{\partial x_i^x} + \mathbf{v}_{ij}^y \frac{\partial W_{ij}}{\partial x_i^y} + \mathbf{v}_{ij}^z \frac{\partial W_{ij}}{\partial x_i^z} \right) \quad (20)$$

2.3.2 Conservación de Momentum

$$\begin{aligned} \frac{Dv_i^x}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{xx}}{\rho_i^2} + \frac{\sigma_j^{xx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\ & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{xy}}{\rho_i^2} + \frac{\sigma_j^{xy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} \\ & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{xz}}{\rho_i^2} + \frac{\sigma_j^{xz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^x \end{aligned} \quad (21)$$

$$\begin{aligned} \frac{Dv_i^y}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{yx}}{\rho_i^2} + \frac{\sigma_j^{yx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\ & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{yy}}{\rho_i^2} + \frac{\sigma_j^{yy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} \\ & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{yz}}{\rho_i^2} + \frac{\sigma_j^{yz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^y \end{aligned} \quad (22)$$

$$\begin{aligned}
\frac{Dv_i^z}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{zx}}{\rho_i^2} + \frac{\sigma_j^{zx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\
& - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{zy}}{\rho_i^2} + \frac{\sigma_j^{zy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} \\
& - \sum_{j=1}^N m_j \left(\frac{\sigma_i^{zz}}{\rho_i^2} + \frac{\sigma_j^{zz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^z
\end{aligned} \tag{23}$$

2.4 Conservación de momento expandida con el tensor de tasa de deformación de corte

$$\begin{aligned}
\frac{Dv_i^x}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{xx}}{\rho_i^2} + \frac{\mu \varepsilon_j^{xx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\
& + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{xy}}{\rho_i^2} + \frac{\mu \varepsilon_j^{xy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{xz}}{\rho_i^2} + \frac{\mu \varepsilon_j^{xz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^x, \\
\frac{Dv_i^y}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{yx}}{\rho_i^2} + \frac{\mu \varepsilon_j^{yx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\
& + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{yy}}{\rho_i^2} + \frac{\mu \varepsilon_j^{yy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{yz}}{\rho_i^2} + \frac{\mu \varepsilon_j^{yz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^y, \\
\frac{Dv_i^z}{Dt} = & - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{zx}}{\rho_i^2} + \frac{\mu \varepsilon_j^{zx}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i} \\
& + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{zy}}{\rho_i^2} + \frac{\mu \varepsilon_j^{zy}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial y_i} + \sum_{j=1}^N m_j \left(\frac{\mu \varepsilon_i^{zz}}{\rho_i^2} + \frac{\mu \varepsilon_j^{zz}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial z_i} + g^z.
\end{aligned}$$

2.4.1 Ecuaciones de la tasa de deformación por corte $\varepsilon^{\alpha\beta}$

$$\begin{aligned}
\varepsilon^{xx} &= \frac{\partial v^x}{\partial x} + \frac{\partial v^x}{\partial x} - \frac{2}{3}(\nabla \cdot \mathbf{v}) \\
\varepsilon^{yy} &= \frac{\partial v^y}{\partial y} + \frac{\partial v^y}{\partial y} - \frac{2}{3}(\nabla \cdot \mathbf{v}) \\
\varepsilon^{zz} &= \frac{\partial v^z}{\partial z} + \frac{\partial v^z}{\partial z} - \frac{2}{3}(\nabla \cdot \mathbf{v}) \\
\varepsilon^{xy} &= \frac{\partial v^x}{\partial y} + \frac{\partial v^y}{\partial x} \\
\varepsilon^{yx} &= \frac{\partial v^y}{\partial x} + \frac{\partial v^x}{\partial y} \\
\varepsilon^{yz} &= \frac{\partial v^y}{\partial z} + \frac{\partial v^z}{\partial y} \\
\varepsilon^{zy} &= \frac{\partial v^z}{\partial y} + \frac{\partial v^y}{\partial z} \\
\varepsilon^{zx} &= \frac{\partial v^z}{\partial x} + \frac{\partial v^x}{\partial z} \\
\varepsilon^{xz} &= \frac{\partial v^x}{\partial z} + \frac{\partial v^z}{\partial x}
\end{aligned} \tag{24}$$

2.4.2 Tasa de deformación viscosa desglosada

$$\varepsilon_i^{\alpha\beta} = \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^\beta \frac{\partial W_{ij}}{\partial x_i^\alpha} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^\alpha \frac{\partial W_{ij}}{\partial x_i^\beta} - \left(\frac{2}{3} \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji} \cdot \nabla_i W_{ij} \right) \delta^{\alpha\beta} \tag{25}$$

Para la partícula j en el método SPH, la ecuación para la tasa de deformación viscosa sería similar, pero intercambiando los índices i y j en todas las cantidades asociadas con la partícula i . Por lo tanto, la expresión para la tasa de deformación viscosa para la partícula j sería:

$$\varepsilon_j^{\alpha\beta} = \sum_{i=1}^N \frac{m_i}{\rho_i} v_{ij}^\beta \frac{\partial W_{ij}}{\partial x_j^\alpha} + \sum_{i=1}^N \frac{m_i}{\rho_i} v_{ij}^\alpha \frac{\partial W_{ij}}{\partial x_j^\beta} - \left(\frac{2}{3} \sum_{i=1}^N \frac{m_i}{\rho_i} v_{ij} \cdot \nabla_j W_{ij} \right) \delta^{\alpha\beta} \tag{26}$$

y por cada componente, para la partícula i es:

$$\begin{aligned}
\varepsilon_i^{xx} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^x} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^x} - \left(\frac{2}{3} \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji} \cdot \nabla_i W_{ij} \right) \delta^{xx} \\
\varepsilon_i^{yy} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^y} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^y} - \left(\frac{2}{3} \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji} \cdot \nabla_i W_{ij} \right) \delta^{yy} \\
\varepsilon_i^{zz} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^z} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^z} - \left(\frac{2}{3} \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji} \cdot \nabla_i W_{ij} \right) \delta^{zz} \\
\varepsilon_i^{xy} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^x} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^y} \\
\varepsilon_i^{yx} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^y} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^x} \\
\varepsilon_i^{yz} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^y} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^z} \\
\varepsilon_i^{zy} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^y \frac{\partial W_{ij}}{\partial x_i^z} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^y} \\
\varepsilon_i^{zx} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^z} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^x} \\
\varepsilon_i^{xz} &= \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^z \frac{\partial W_{ij}}{\partial x_i^x} + \sum_{j=1}^N \frac{m_j}{\rho_j} v_{ji}^x \frac{\partial W_{ij}}{\partial x_i^z}
\end{aligned} \tag{27}$$

Si se tiene la forma matricial de $\varepsilon_i^{\alpha\beta}$, se puede realizar la operación $\varepsilon_i^{\alpha\beta} \varepsilon_i^{\alpha\beta}$ expandiendo y sumando los elementos correspondientes. La forma matricial de $\varepsilon_i^{\alpha\beta}$ podría ser expresada como una matriz D de dimensiones 3×3 para cada partícula i , donde α y β toman valores x , y y z . Así, para una partícula dada, $\varepsilon_i^{\alpha\beta}$ se representa como:

$$D = \begin{bmatrix} \varepsilon_i^{xx} & \varepsilon_i^{xy} & \varepsilon_i^{xz} \\ \varepsilon_i^{yx} & \varepsilon_i^{yy} & \varepsilon_i^{yz} \\ \varepsilon_i^{zx} & \varepsilon_i^{zy} & \varepsilon_i^{zz} \end{bmatrix} \tag{28}$$

La operación $\varepsilon_i^{\alpha\beta} \varepsilon_i^{\alpha\beta}$ implica tomar el producto punto de la matriz D_i consigo misma, sumando los cuadrados de todos los elementos:

$$\varepsilon_i^{\alpha\beta} \varepsilon_i^{\alpha\beta} = \sum_{\alpha\beta} (D_i^{\alpha\beta})^2 \tag{29}$$

Expandiendo esta expresión para cada componente, obtenemos:

$$\begin{aligned} \varepsilon_i^{\alpha\beta} \varepsilon_i^{\alpha\beta} = & (\varepsilon_i^{xx})^2 + (\varepsilon_i^{xy})^2 + (\varepsilon_i^{xz})^2 + \\ & (\varepsilon_i^{yx})^2 + (\varepsilon_i^{yy})^2 + (\varepsilon_i^{yz})^2 + \\ & (\varepsilon_i^{zx})^2 + (\varepsilon_i^{zy})^2 + (\varepsilon_i^{zz})^2 \end{aligned} \quad (30)$$

Este resultado representa la magnitud cuadrada de la tasa de deformación por corte para una partícula i específica en función de sus componentes individuales.

3 Funciones de Suavizado

Las funciones de suavizado deben cumplir con las características detalladas en la Sección 2. Además, se requiere que la función de suavizado tenga una segunda derivada continua y que tanto el núcleo como sus derivadas sean computacionalmente eficientes.

Hay diversas opciones disponibles para la función de suavizado. No obstante, la elección del núcleo tiene un impacto significativo en la precisión de las aproximaciones a las propiedades físicas de las partículas.

3.1 Núcleo Gaussiano (Gaussian kerne)

El núcleo gaussiano es lo suficientemente suave incluso para órdenes altos de derivadas, y es muy estable y preciso, especialmente para partículas desordenadas. Sin embargo no es compacto, lo que implica que su dominio sea muy extendido lo que resulta en un alto costo computacional. El nucleo Gaussiano está dado por:

$$W(R, h) = \alpha_d e^{-R^2}, \quad (31)$$

donde α_d es $1/\pi^{1/2}h$, $1/\pi h^2$, y $1/\pi^{3/2}h^2$, para una, dos y tres dimensiones.

3.2 Núcleo de Spline Cúbico (Cubic Spline Kernel):

El núcleo de spline cúbico es ampliamente preferido debido a su comportamiento matemático y a la eficiencia computacional de sus derivadas (Liu & Liu 2010). Esta dado por:

$$W(x - x') = W(R, h) = \alpha_d \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3, & 0 \leq R < 1 \\ \frac{1}{6}(2 - R)^3, & 1 \leq R < 2 \\ 0, & R \geq 2, \end{cases} \quad (32)$$

donde α_d es $1/h$, $15/7\pi h^2$, y $3/2\pi h^3$, para una, dos y tres dimensiones, respectivamente. Este núcleo es uno de los más usados debido a que es más parecido a la función Gaussiana, más angosta y compacta. Sin embargo, la segunda derivada es una función lineal por partes, y según las propiedades de estabilidad puede ser menos suave que otras funciones de núcleo.

3.3 Núcleo de Spline Quintic (Quintic Spline Kernel):

$$W(x - x') = W(R, h) = \alpha_d \begin{cases} (3 - R)^5 - 6(2 - R)^5 + 15(1 - R)^5, & 0 \leq R < 1, \\ (3 - R)^5 - 6(2 - R)^5, & 1 < R \leq 2, \\ (3 - R)^5, & 2 < R \leq 3, \\ 0, & R > 3, \end{cases} \quad (33)$$

donde α_d es $120/h$, $7/478\pi h^2$ y $359/208\pi h^3$ para una, dos y tres dimensiones respectivamente.

4 Búsqueda de partículas vecinas

Hay diferentes métodos desarrollados en el método SPH, la cual tiene directa influencia en el tiempo de computación. En dos dimensiones las listas de vecinos (enlazadas y de Verlet) son comunmente en simulaciones de partículas (Domínguez et al. 2011; Viccione et al. 2008). Para simulaciones en tres dimensiones es usada la técnica del octree (Rajasekaran & Reif 2007).

En el método directo, la partícula i es fijada, y todos los pares de partículas ij dentro del dominio serán buscadas mediante la comparación de sus distancias, si es menor que el radio de soporte del dominio, kh , la pareja de partículas será guardada en la lista.

4.1 Método Grid Hashing

Gross et al. (2019) demuestra que el método de grid hashing permite una búsqueda eficiente de vecinos al dividir el espacio en una cuadrícula y utilizar funciones hash para asignar elementos a celdas específicas. Este enfoque reduce considerablemente el número de comparaciones necesarias, lo que mejora el rendimiento en aplicaciones de búsqueda de vecinos más cercanos. Este tipo de métodos ha sido previamente implementado en simulaciones computacionales de SPH aceleradas por GPU (Goswami et al. 2010).

En este trabajo, se emplea el método de grid hashing para realizar la búsqueda de vecinos, logrando una mejora significativa en la rapidez del código, incrementando su velocidad entre 5 y 10 veces en comparación con el enfoque de búsqueda directa de vecinos.

En la Figura 2, se presenta una representación gráfica de un conjunto de partículas distribuidas aleatoriamente en un dominio bidimensional, con coordenadas x e y . El espacio de simulación se ha dividido en una cuadrícula de 3×3 , donde cada celda ha sido numerada del 1 al 9, lo que permite identificar las zonas del dominio de manera sistemática.

En el centro de la cuadrícula (celda 5), se seleccionó un punto particular, el cual se marca en rojo. Alrededor de este punto, se trazó un círculo de radio kh , que representa el radio de influencia o dominio de interacción utilizado en los cálculos del método de Smoothed Particle Hydrodynamics (SPH). Este radio determina la vecindad en la que se evalúan las interacciones entre partículas para el cálculo de las fuerzas y propiedades físicas del sistema.

Las partículas ubicadas dentro de esta región de influencia (es decir, aquellas cuya distancia al punto central es menor o igual a kh se resaltan en color rojo, mientras que las partículas fuera de este dominio se mantienen en color azul. La línea negra que se extiende desde el punto central hasta el borde del círculo ilustra el valor de kh , que es un parámetro clave para determinar el alcance de las interacciones en simulaciones SPH.

Esta representación permite visualizar cómo se define la vecindad de un punto en un contexto de simulación de dinámica de fluidos, donde solo las partículas dentro del radio de influencia contribuyen a los cálculos de las ecuaciones gobernantes del sistema. La división del espacio en una cuadrícula también facilita el análisis de la distribución espacial de las partículas y la identificación de regiones específicas de interés en el dominio.

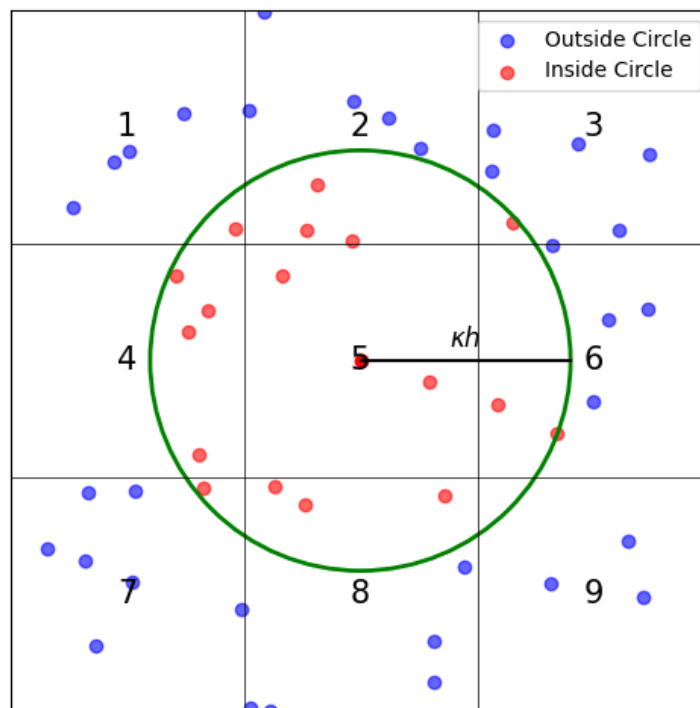


Figure 2: La figura permite visualizar cómo se define la vecindad de un punto en un contexto de simulación de dinámica de fluidos, donde solo las partículas dentro del radio de influencia contribuyen a los cálculos de variables para las ecuaciones gobernantes.

5 Método de salto de rana

El método de salto de rana, también conocido como Leapfrog en inglés, es un integrador numérico ampliamente empleado para resolver ecuaciones diferenciales ordinarias de segundo orden. Aquí se presenta una descripción detallada de cómo se aplica este método a ecuaciones diferenciales de la forma $\frac{d^2y}{dt^2} = f(y, t)$:

1. **Inicialización:** Se inicia el proceso con las condiciones iniciales y_0 y $v_0 = \frac{dy}{dt}$ en el tiempo inicial t_0 .
2. **Paso de medio paso:** Se calcula el valor de v en el paso de medio paso adelante, utilizando el valor de y en el tiempo actual y la derivada $f(y, t)$:

$$v_{n+1/2} = v_n + \frac{1}{2} \cdot f(y_n, t_n) \cdot \Delta t$$

3. **Paso de posición completa:** Se calcula el valor de y en el próximo paso de tiempo completo, utilizando el valor de v en el paso de medio paso adelante:

$$y_{n+1} = y_n + v_{n+1/2} \cdot \Delta t$$

4. **Paso de aceleración completa:** Se calcula la derivada $f(y, t)$ utilizando el nuevo valor de y_{n+1} en el tiempo t_{n+1} :

$$f_{n+1} = f(y_{n+1}, t_{n+1})$$

5. **Paso final de velocidad completa:** Se calcula el valor de v en el próximo paso de tiempo completo, utilizando la derivada actualizada f_{n+1} :

$$v_{n+1} = v_{n+1/2} + \frac{1}{2} \cdot f_{n+1} \cdot \Delta t$$

6. **Avance de tiempo:** Se avanza al siguiente paso de tiempo $t_{n+1} = t_n + \Delta t$ y se repite el proceso desde el paso 2.

El método de salto de rana es de segundo orden, lo que significa que el error en la posición y la velocidad son proporcionales a $(\Delta t)^2$. Se destaca por su estabilidad y capacidad para conservar la energía en sistemas hamiltonianos, lo que lo convierte en una herramienta invaluable en simulaciones de sistemas físicos.

6 Modelo reológico

Los modelos reológicos son representaciones matemáticas que describen el comportamiento de los fluidos bajo el esfuerzo de cizalla. Estos modelos ayudan a entender cómo diferentes tipos de materiales responden a la aplicación de una fuerza, lo cual es esencial en áreas como la ingeniería de suelos y la dinámica de flujos complejos como los lodos.

6.1 Modelo de Herschel-Bulkley

El modelo de Herschel-Bulkley es una generalización del modelo de Bingham, que permite un comportamiento no lineal entre el esfuerzo cortante y la tasa de deformación. La ecuación está dada por:

$$\tau = \tau_y + K\dot{\gamma}^n$$

donde:

- τ es el esfuerzo cortante,
- τ_y es el umbral de fluencia,
- K es el coeficiente de consistencia,
- $\dot{\gamma}$ es la tasa de deformación,
- n es el índice de comportamiento del flujo.

Este modelo es adecuado para lodos que muestran un comportamiento viscoplástico con un componente no lineal en su flujo, como algunos flujos de lodo con contenido variable de sólidos.

Los fluidos no newtonianos pueden exhibir un comportamiento dilatante, en el que la viscosidad aparente del fluido aumenta con la velocidad de cizalla. La figura 3 muestra cómo varía la fuerza de cizalla con la velocidad de cizalla al modificar el valor del índice n . Las curvas se generaron para valores de n que van desde 1 hasta 2, incrementándose en pasos uniformes. Este análisis permite observar que, a medida que el valor de n aumenta, la pendiente de la curva se incrementa, indicando un aumento más rápido de la fuerza de cizalla con respecto a la velocidad de cizalla.

La figura 4 muestra la dependencia de la fuerza de cizalla al variar el coeficiente de consistencia K , manteniendo un índice fijo de $n = 1.8$. Se observará que, para valores más altos de K , el fluido presenta una mayor resistencia al movimiento, lo que se traduce en una mayor fuerza de cizalla para la misma velocidad de cizalla. Esto refleja que el coeficiente K está directamente relacionado con la viscosidad del fluido.

Las gráficas obtenidas y analizadas muestran cómo los parámetros n y K afectan el comportamiento de los fluidos dilatantes. Un valor mayor de n intensifica la dependencia no lineal entre la fuerza de cizalla y la velocidad de cizalla, mientras que un aumento en K incrementa la viscosidad general del fluido, resultando en una mayor fuerza de cizalla para la misma velocidad de cizalla.

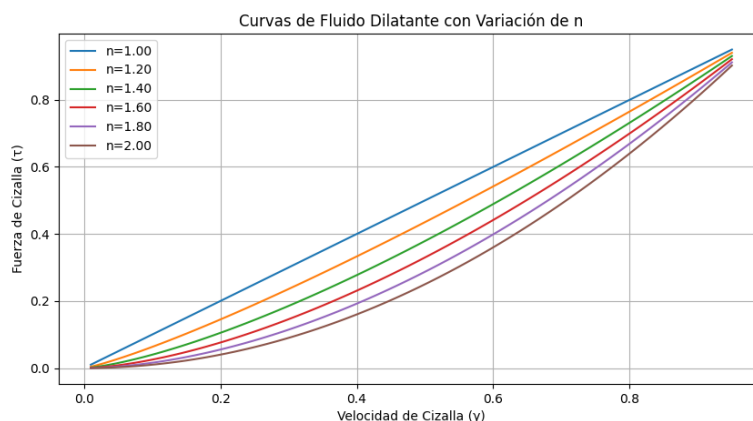


Figure 3: Curvas de comportamiento dilatante para diferentes valores del índice n (1 a 2).

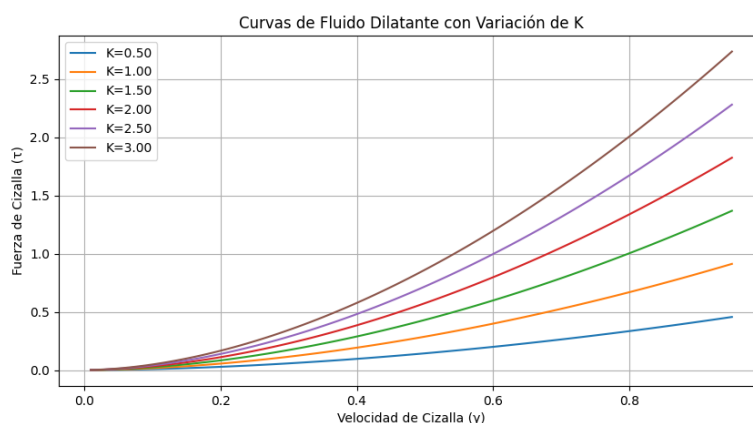


Figure 4: Curvas de comportamiento dilatante para diferentes valores de K con $n = 1.8$.

6.2 Modelos Reológicos (Herschel-Bulkley y Bingham)

Dependiendo del tipo de material y su concentración, se han propuesto modelos como el de Herschel-Bulkley para describir el comportamiento del flujo. Este modelo ajusta bien a materiales que exhiben comportamiento viscosoplástico, con índices de potencia (n) que varían con la composición.

Suspensiones con partículas de arcilla tienden a comportarse de forma consistente con un índice de potencia bajo, mientras que materiales de grano más grueso (como arena) muestran índices de potencia más altos.

El modelo de Bingham, que asume una viscosidad constante, solo es adecuado para describir estos materiales a tasas de corte muy altas (mayores a $10\text{--}20\text{ s}^{-1}$), mientras que para tasas bajas o moderadas de corte, como las de flujos de escombros, su precisión es limitada. El modelo de fluido de Bingham se expresa como:

$$\tau = \tau_0 + K \cdot \dot{\gamma} \quad (34)$$

donde τ_0 es la tensión de fluencia. Este modelo es característico de materiales que no comienzan a fluir hasta que se supera un cierto umbral de esfuerzo (τ_0), como algunos lodos de alta viscosidad.

En el modelo de Herschel-Bulkley, el valor del índice de potencia n describe el tipo de comportamiento reológico de diferentes materiales en función de la composición de la suspensión. En resumen:

- **Modelo de fluido newtoniano:** se describe mediante la siguiente ecuación:

$$\tau = K \cdot \dot{\gamma}$$

donde η es la viscosidad del fluido y $n = 1$. Este comportamiento es lineal y representa fluidos como el agua o el aire.

- **Fluido Pseudoplástico:** Los fluidos pseudoplásticos, o de adelgazamiento por cizalla, se comportan de la siguiente forma:

$$\tau = K \cdot \dot{\gamma}^n \quad \text{con } 0 < n < 1$$

Este modelo describe materiales como suspensiones de arcilla y algunos lodos con partículas finas, que disminuyen su viscosidad a medida que aumenta la velocidad de cizalla.

- *Fluidos con partículas finas (como arcilla):* Índice de potencia bajo, típicamente alrededor de $n = 0.33$. Esto representa un comportamiento más pseudoplástico, donde el fluido se vuelve menos viscoso a medida que aumenta la tasa de corte. Este tipo de comportamiento es común en suspensiones con partículas muy pequeñas y cohesivas que fluyen con facilidad bajo esfuerzo.
- *Fluidos con partículas más gruesas (como arena fina):* índice de potencia intermedio, entre $n = 0.75$ y $n = 0.92$. Estos materiales muestran una viscosidad que disminuye con el esfuerzo, pero en menor medida que las suspensiones de arcilla. El comportamiento sigue siendo pseudoplástico, aunque menos pronunciado.
- **Fluido Dilatante:** el modelo dilatante, o de engrosamiento por cizalla, se describe por:

$$\tau = K \cdot \dot{\gamma}^n \quad \text{con } n > 1$$

Este tipo de comportamiento se observa en suspensiones con partículas gruesas como arena, donde la viscosidad aumenta con la velocidad de cizalla.

- *Fluidos con partículas aún más gruesas (como arena más gruesa):* índice de potencia alto, que puede llegar hasta $n = 1.35$. Este valor representa un comportamiento más dilatante, donde la viscosidad aumenta con la tasa de corte. Esto ocurre en suspensiones de partículas grandes que tienden a interactuar y colisionar más intensamente, aumentando la resistencia al flujo a medida que se incrementa el esfuerzo.

En general, a medida que el tamaño de partícula aumenta en la suspensión, el índice de potencia n también tiende a aumentar, reflejando un cambio de comportamiento de pseudoplástico a dilatante en el material. La siguiente figura muestra las curvas para ilustrar estos modelos reológicos:

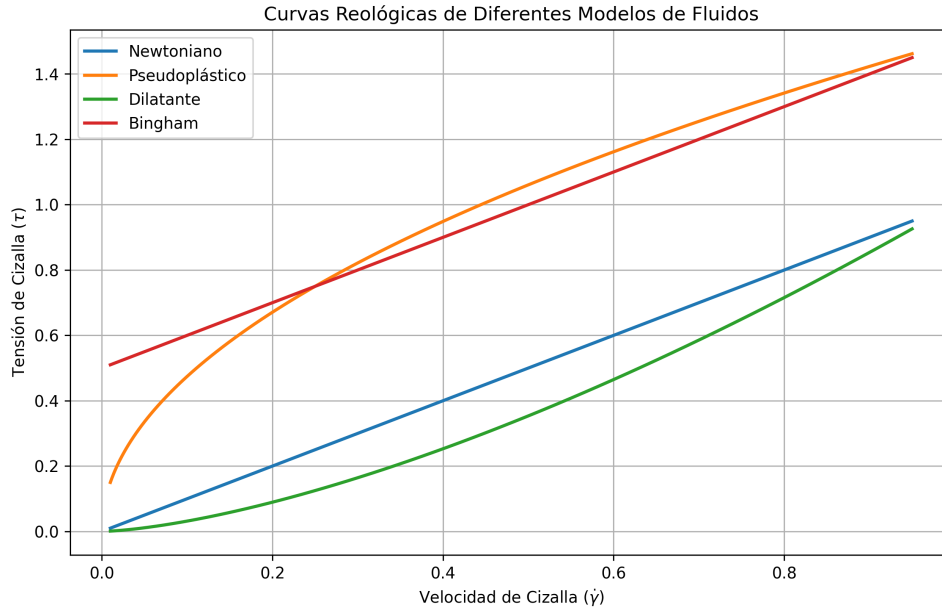


Figure 5: Curvas de tensión de cizalla (τ) frente a la velocidad de cizalla ($\dot{\gamma}$) para diferentes tipos de fluidos.

En resumen, el comportamiento del flujo de lodos puede variar ampliamente con pequeños cambios en la composición del material, y modelos reológicos avanzados como Herschel-Bulkley se utilizan para capturar el comportamiento de estos materiales bajo diferentes condiciones de flujo.

6.2.1 Formulación tensorial - SPH del modelo de Herschel-Bulkley

En el marco de modelado de fluidos no newtonianos, la tensión cortante se define mediante:

$$\tau^{\alpha\beta} = K \left(\frac{1}{2} \varepsilon_{\gamma\delta} \varepsilon^{\gamma\delta} \right)^{\frac{n-1}{2}} \varepsilon^{\alpha\beta}, \quad (35)$$

donde:

- K es la consistencia del fluido, una constante dependiente de la naturaleza del material.
- n es el índice de comportamiento reológico:

- $n > 1$: fluido dilatante, donde la viscosidad aumenta con la tasa de deformación.
- $n < 1$: fluido pseudoplástico, donde la viscosidad disminuye con la tasa de deformación.
- $n = 1$: fluido newtoniano, donde la viscosidad es constante e independiente de $\dot{\gamma}$.
- $\varepsilon^{\alpha\beta}$ es el tensor de tasa de deformación, y $\varepsilon_{\gamma\delta}\varepsilon^{\gamma\delta}$ representa el segundo invariante de dicho tensor.

Ventajas de la formulación unificada

La definición de $\tau^{\alpha\beta}$ presentada anteriormente posee varias ventajas que justifican su implementación en modelos de dinámica de fluidos:

1. Generalidad y flexibilidad

La formulación unificada permite describir una amplia gama de comportamientos reológicos:

- En fluidos newtonianos ($n = 1$), $\tau^{\alpha\beta} = K\varepsilon^{\alpha\beta}$, lo que implica una viscosidad constante.
- En fluidos no newtonianos ($n \neq 1$), la viscosidad varía con la tasa de deformación de manera proporcional al índice n .

2. Dependencia escalar de la tasa de deformación

El término $\varepsilon_{\gamma\delta}\varepsilon^{\gamma\delta}$ es una magnitud escalar que relaciona directamente la tensión cortante con la tasa de deformación efectiva:

$$\dot{\gamma} = \sqrt{\frac{1}{2}\varepsilon_{\gamma\delta}\varepsilon^{\gamma\delta}}.$$

Esto proporciona una interpretación física intuitiva y consistente, ya que el comportamiento del fluido depende de las propiedades invariantes del tensor de deformación.

3. Consistencia matemática

La formulación mantiene la simetría y las propiedades tensoriales del esfuerzo cortante. Además, garantiza que $\tau^{\alpha\beta}$ siga el mismo eje tensorial que $\varepsilon^{\alpha\beta}$, lo que preserva la coherencia entre las fuerzas viscosas y los gradientes de velocidad.

4. Adaptabilidad numérica

En simulaciones numéricas, esta formulación simplifica el cálculo de la viscosidad efectiva:

$$\eta_{\text{eff}} = K\dot{\gamma}^{n-1}.$$

Esto evita ajustes ad hoc para diferentes fluidos, ya que K y n encapsulan toda la variabilidad reológica.

5. Interpretación física directa

El término $(0.5\varepsilon_{\gamma\delta}\varepsilon^{\gamma\delta})^{\frac{n-1}{2}}$ modula la viscosidad efectiva del fluido según la tasa de deformación. Esto permite capturar:

- Aumento de resistencia al flujo en fluidos dilatantes ($n > 1$).
- Disminución de resistencia en fluidos pseudoplásticos ($n < 1$).
- Resistencia constante en fluidos newtonianos ($n = 1$).

Implicaciones de la formulación

Esta formulación unificada no solo permite modelar fluidos complejos, sino que también asegura:

- Una descripción coherente del comportamiento viscoso para diferentes índices n .
- Facilidad de implementación en métodos numéricos como SPH, gracias a la dependencia clara de magnitudes escalares.
- Consistencia matemática y física en la representación del esfuerzo cortante.

Por lo tanto, la implementación de $\tau^{\alpha\beta} = K (\varepsilon_{\gamma\delta}\varepsilon^{\gamma\delta})^{\frac{n-1}{2}} \varepsilon^{\alpha\beta}$ es esencial para capturar el comportamiento de fluidos no newtonianos en un marco generalizado.

7 Descripción de Rutinas

La implementación de las rutinas está disponible en un repositorio de GitHub:

<https://github.com/diegovalencia82/massmove.git>

A continuación, se describen estas subrutinas.

7.1 Rutina sph

Esta rutina constituye el núcleo principal de un programa básico en Fortran diseñado para un código de simulación hidrodinámica de partículas suavizadas (SPH) en tres dimensiones, integrando y gestionando las demás rutinas asociadas.

Variables

El programa se estructura en torno a un espacio de memoria multidimensional, `mspace`, que sirve para almacenar las propiedades de las partículas. Las variables empleadas en el programa se detallan en la siguiente tabla:

Table 1: Parámetros de entrada utilizados en la rutina.

Variables	Tipo	Descripción	In / out
<code>mspace(26, nmax)</code>	<code>double precision</code>	Matriz bidimensional que almacena información de las partículas, donde: <ul style="list-style-type: none"> Primera dimensión (26) corresponde a diferentes propiedades físicas. Segunda dimensión <code>nmax</code> define el número máximo de partículas permitido. 	[in/out]
<code>dt</code>	<code>double precision</code>	Paso de tiempo utilizado en la integración.	[in/out]
<code>ntotal</code>	<code>integer</code>	Número total de partículas del modelo.	[out]
<code>nfluid</code>	<code>integer</code>	Número total de partículas de fluido.	[in]
<code>nvirt</code>	<code>integer</code>	Número total de partículas de frontera o paredes.	[in]

<code>ntype(2)</code>	<code>integer</code>	Matriz que almacena el numero de partículas en el transcurso de la simulación. <code>ntype(1)</code> : el número de partículas total y <code>ntype(2)</code> : el número de partículas de frontera.	[in/out]
<code>start_clock</code>	<code>integer</code>	Tiempo de inicio de la simulación	[out]
<code>end_clock</code>	<code>integer</code>	Tiempo de final de la simulación	[out]
<code>rate</code>	<code>integer</code>	Tasa de cambio	[out]

La variable `mspace(26, nmax)` define una Matriz bidimensional común que almacena información de las partículas, donde:

- Primera dimensión (26) corresponde a diferentes propiedades físicas.
- Segunda dimensión (`nmax`) define el número máximo de partículas permitido.

Cada fila de `mspace` tiene una función específica como se muestra en la tabla 2:

Table 2: Parámetros de entrada utilizados en la rutina.

Matriz	Variable	Descripción
<code>mspace(1,i)</code>	(id)	Etiqueta o número de la partícula
<code>mspace(2 - 4,i)</code>	(x, y, z)	Posiciones espaciales de las partículas. El espacio y o 3 se deshabilita en 2D.
<code>mspace(5 - 7,i)</code>	(v_x, v_y, v_z)	Velocidades en cada dirección. El espacio v_y o 6 se deshabilita en 2D.
<code>mspace(8,i)</code>	m	Masa de cada partícula.
<code>mspace(9,i)</code>	ρ	Densidad del fluido asociada a la partícula.
<code>mspace(10,i)</code>	p	Presión calculada para la partícula.
<code>mspace(11,i)</code>	u	Energía interna asociada a la partícula. Deshabilitada.
<code>mspace(12,i)</code>	(itype)	Identifica si la partícula es tipo fluido o frontera.
<code>mspace(13,i)</code>	h	Longitud de suavizado.
<code>mspace(14,i)</code>	c	Velocidad del sonido.
<code>mspace(15,i)</code>	s	Entropía. Deshabilitada.

Matriz	Variable	Descripción
<code>mspace(16,i)</code>	e	Energía total de las partículas. Deshabilitada.
<code>mspace(17,i)</code>	Distancia $r_{fi} - r_{vi}$	Distancia entre partícula de fluido f y partícula de frontera v .
<code>mspace(18,i)</code>	parametro de viscosidad K	Parámetro calculado dependiente de la concentración.
<code>mspace(19,i)</code>	Parámetro n	Exponente del modelo de Herschel-Bulkley.
<code>mspace(20 - 22,i)</code>	(Dv_x, Dv_y, Dv_z)	Cálculo de derivadas de velocidad. En 2D se deshabilita Dv_y .
<code>mspace(21 - 24,i)</code>	Espacio para cálculos	Deshabilitada.
<code>mspace(25,i)</code>	$D\rho$	Derivada de la densidad. Deshabilitada.
<code>mspace(26,i)</code>	K	Cálculo de coeficiente de viscosidad

Flujo de la rutina

En primer lugar, el programa carga las propiedades iniciales de todas las partículas en `mspace`. A continuación, ejecuta la integración temporal, resolviendo las ecuaciones de movimiento para actualizar las posiciones, velocidades, densidades y otras propiedades. Finalmente, registra el tiempo total de ejecución. Las operaciones principales que realiza el programa son las siguientes:

1. Declaración de variables:

- **ntotal**: Número total de partículas.
- **nfluid** y **nvirt**: Número de partículas fluidas y virtuales.
- **ntype(2)**: Contador para fluidos y partículas virtuales.
- **dt**: Paso de tiempo para la integración.
- **mspace**: Matriz principal de datos (definida en el módulo).

2. Medición del tiempo de ejecución:

- Se usa la función **CPU_TIME** para medir el tiempo de inicio y fin del programa.

3. Inicialización:

- **call input**: Se inicializan los datos de `mspace` (propiedades de las partículas) junto con el número de partículas totales y el tipo de partículas (**nfluid**, **nvirt**).

4. Configuración: Define los tipos de partículas:

- `ntype(1)` contiene todas las partículas (fluidas y virtuales).
- `ntype(2)` almacena solo las partículas virtuales.

5. Integración temporal:

- **call `time_integration`:** Llama a la rutina principal de integración para calcular el estado del sistema en pasos de tiempo sucesivos.

6. Calculo de eficiencia:

- Almacena los tiempos de CPU en el archivo `time_CPU.dat`, incluyendo tiempo inicial, final y tiempo total de ejecución en horas.

7.2 Rutina input

La rutina `input` se encarga de leer y cargar la configuración inicial de las partículas de un archivo de entrada en la matriz `mspace`.

Variables

Table 3: Parámetros de entrada utilizados en la rutina `input`.

Variables	Tipo	Descripción	In / out
<code>mspace(26, nmax)</code>	<code>double precision</code>	Matriz bidimensional que almacena información de las partículas.	[in/out]
<code>time</code>	<code>double precision</code>	tiempo en segundos	[in]
<code>ntotal</code>	<code>integer</code>	Número total de partículas del modelo.	[in]
<code>nfluid</code>	<code>integer</code>	Número total de partículas de fluido.	[in]
<code>nvirt</code>	<code>integer</code>	Número total de partículas de frontera o paredes.	[in]
<code>itimestep</code>	<code>integer</code>	Paso de tiempo inicial	[in]

Flujo de la rutina

La rutina extrae información relevante sobre las partículas, como posiciones, velocidades, masas, densidades, entre otras propiedades, desde un archivo externo denominado `snapshot_000`. Este archivo corresponde a la instantánea inicial del sistema y sirve como punto de partida para los cálculos. Además, los archivos de salida generados para cada instantánea siguen un formato similar al de la tabla 4.

Los datos iniciales se extraen de un archivo como el mostrado en la tabla 4 y se almacenan en la matriz multidimensional `mspace`, que sirve como base para los cálculos

Table 4: Formato de archivo inicial y de salida.

itimestep			time			ntotal		nfluid			nvirt				
id_1	x_1	y_1	z_1	vx_1	vy_1	vz_1	m_1	ρ_1	p_1	u_1	$itype_1$	h_1	K_1		
id_2	x_2	y_2	z_2	vx_2	vy_2	vz_2	m_2	ρ_2	p_2	u_2	$itype_2$	h_2	K_2		
...
...

posteriores. Esta estructura permite manejar las propiedades de las partículas de manera eficiente durante las simulaciones.

7.3 Rutina time_integration

La subrutina `time_integration` implementa la integración temporal del movimiento de partículas en un sistema de dinámica de fluidos utilizando el método de salto de rana (*Leapfrog*). Además, incluye la funcionalidad de inyección de nuevas partículas desde una fuente, lo cual permite simular flujos con incorporación continua de masa.

Variables

Table 5: Parámetros de entrada utilizados en la rutina `input`.

Variables	Tipo	Descripción	In / out
<code>mspace(26, nmax)</code>	double precision	Matriz bidimensional que almacena información de las partículas actualizada en cada paso de iteración.	[in/out]
<code>ntype</code>	integer	Número de partículas tipo fluidos y frontera.	[in]
<code>dt</code>	double precision	Paso de tiempo.	[in]
<code>infilebas</code>	character	Archivo de datos inicial	[in]
<code>outfile</code>	character	Archivo de datos de salida	[out]

Flujo General

El flujo general de la subrutina es el siguiente:

1. **Inicialización:** Se inicializan variables necesarias para la simulación, incluyendo los parámetros geométricos y físicos asociados a la fuente de partículas, tales como velocidad de emisión, volumen emitido, masa de partículas, y pasos de tiempo controlados por la condición de Courant.

2. **Emisión de partículas:** La subrutina se adapta dinámicamente al número de partículas mediante el control de índices en la matriz `mspace`. La configuración de las partículas generadas depende de las dimensiones espaciales (`h0`) y del flujo volumétrico. Además, se proporciona compatibilidad para simulaciones con múltiples tipos de partículas y condiciones iniciales.

- Cada cierto número de pasos de tiempo (`nstepsf`), controlado por la frecuencia de emisión, se calcula el número de partículas a generar en la fuente según el flujo volumétrico (`QQ`) y se posicionan en una cuadrícula definida por la geometría de la fuente.
- Para cada capa de partículas, se llama a

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(r_{ij}, h_i) + \mathbf{g},$$

donde \mathbf{v}_i es la velocidad de la partícula i , \mathbf{g} es la aceleración gravitacional, y ∇W es el gradiente de la función de suavizado. la subrutina `fuelle`, la cual actualiza el estado del sistema incluyendo las nuevas partículas en las estructuras de datos.

3. **Cálculo de propiedades dinámicas:**

- Se actualizan las velocidades y posiciones de las partículas utilizando las ecuaciones de movimiento mediante la rutina `single_step`. La integración temporal sigue el método de salto de rana, el cual divide el cálculo en dos pasos:
 - Actualización de las velocidades en un paso intermedio.
 - Actualización de las posiciones utilizando las velocidades intermedias.
4. **Almacenamiento de datos:** una vez realizada la integración se almacena los datos en archivo llamado `snashot_ddd` cada ciertos pasos de tiempo; esto se establece desde el inicio de la simulación. El formato de estos archivos es como en el formato mostrado en la condición inicial.

7.4 Rutina fuente

La rutina `fuelle` se encarga de generar partículas en un modelo SPH (Smoothed Particle Hydrodynamics) como una fuente que añade puntos al sistema con una configuración inicial de posición, velocidad, masa, y propiedades físicas específicas. Este procedimiento es esencial para simular dinámicas de fluidos en sistemas en los que se agregan partículas de manera continua, como flujos inyectados o fuentes controladas.

Variabes

Table 6: Parámetros de entrada utilizados en la rutina **fuelle**.

Variables	Tipo	Descripción	In / out
vel	double precision	Velocidad inicial de las partículas generadas.	[in]
masa_particula	double precision	Masa asignada a cada partícula generada.	[in]
num_points	integer	Número de partículas que se generan.	[in]
ntype	integer array	Tipos de partículas en el sistema, donde <code>ntype(1)</code> representa el total y <code>ntype(2)</code> las de frontera.	[in/out]
mspace	double precision, dimension(26, nmax)	Matriz que contiene las propiedades físicas de todas las partículas, como posición, velocidad, masa, densidad, presión, etc.	[in/out]
posy	double precision	Coordenada fija en el eje Y para las partículas generadas.	[in]

Flujo de la rutina

- Inicialización y memoria:** Se asigna memoria para matrices auxiliares y se calculan constantes físicas necesarias para los parámetros de presión y velocidad del sonido.
- Generación de partículas:** Utilizando la posición inicial (`start_x`, `start_z`), el ángulo de inyección (`angle`), y el número de puntos (`num_points`), se calculan las coordenadas 3D de las partículas generadas. Todas las partículas se asignan a una posición fija en el eje Y, determinada por `posy`.

La rutina utiliza una serie de variables locales y globales para calcular y actualizar propiedades de las partículas. Algunas de las más relevantes son:

- **Variables físicas constantes:**
 - `c`: Velocidad del sonido, calculada a partir de constantes físicas.
 - `b`: Parámetro relacionado con la presión inicial y la densidad.
 - **Matrices auxiliares:**
 - `mspaca`: Matriz temporal que almacena una copia de `mspace` para realizar actualizaciones.
 - **Coordenadas de las partículas:**
 - `x_coords`, `y_coords`, `z_coords`: Vectores para las posiciones generadas.
- Asignación de propiedades:** Cada partícula generada se añade a la matriz `mspace` con las siguientes propiedades:

- **Posición inicial:** `x_coords`, `y_coords`, `z_coords`.
- **Velocidad inicial:** Se calcula usando la magnitud `vel` y el ángulo de inyección `angle`.
- **Masa:** Determinada por el valor de entrada `masa_particula`.
- **Presión y densidad:** Calculadas en función de las constantes `rho0` (densidad inicial) y `gamma`.
- **Otros parámetros:** Energía interna, tipo de partícula, y longitud de suavizado.

4. Actualización del sistema:

- Se ajusta el número total de partículas en `ntype(1)`.
- Las nuevas partículas se añaden al final de `mspace`.

5. **Liberación de memoria:** La memoria asignada para la matriz auxiliar `mspaca` se libera al final de la rutina para evitar fugas de memoria.

7.5 Rutina `single_step`

La rutina `single_step` es responsable de calcular el lado derecho de una ecuación diferencial en un solo paso de tiempo, permitiendo la integración temporal de las partículas en un modelo SPH (Smoothed Particle Hydrodynamics). Esta rutina invoca internamente una serie de subrutinas para llevar a cabo los cálculos necesarios. Las cuales son: entrada de datos, búsqueda de vecinos, cálculo de la función de suavizado, cálculo de la densidad, cálculo de la presión, cálculo de la ecuación de momento. En resumen, esta rutina, a través de las subrutinas internas, lleva a cabo todos los cálculos necesarios para realizar la integración por partícula.

Variables

Table 7: Parámetros de entrada utilizados en la rutina `single_step`.

Variables	Tipo	Descripción	In / out
<code>mspace</code>	<code>array</code>	Arreglo multidimensional que almacena propiedades de las partículas, como coordenadas, velocidades, densidad, etc.	[in/out]
<code>ntype</code>	<code>integer</code>	Número de tipos de partículas en la simulación.	[in]
<code>itimestep</code>	<code>integer</code>	Número del paso de tiempo actual.	[in]
<code>dt</code>	<code>double precision</code>	Paso de tiempo para la integración numérica.	[in]

Flujo de la rutina

El propósito de la subrutina **single_step** es realizar la integración de un paso del tiempo de un sistema de partículas utilizando algoritmos de dinámica de partículas suavizadas (SPH). Se calcula el lado derecho de una ecuación diferencial en un solo paso, incluyendo propiedades físicas como densidad, presión y velocidades.

Esta rutina inicializa las variables y parámetros requeridos para el cálculo. Se leen los datos iniciales de las partículas, como sus coordenadas, velocidades, densidades, presiones, energía interna, entre otros. La rutina primero realiza una búsqueda de vecinos mediante la rutina **neighboring_search** que hace una búsqueda directa, o la rutina **neighboring_search_P** que hace una búsqueda directa paralelizada, o la rutina **neighboring_Grid.Hashing** que hace una búsqueda mediante el método Grid Hashing o la rutina **neighboring_Grid.Hashing_P** que hace una búsqueda mediante el método Grid Hashing paralelizado. En estas rutinas se generan las variables **nparis**, **nfilas**, **mvrij**, **mviij**, **mxij**, **mvxij**, que son descritas en la tabla 8. Con todos los pares de interacciones encontrados se procede a calcular las funciones de suavizado y sus derivadas en la rutina **wijdwij**. En este punto ya se tiene listo todas las interacciones para calcular las densidades de cada partícula en la rutina **density**, luego la presión mediante la ecuación de estado en la rutina **presioni** y finalmente la ecuación de momento en la rutina **momento**.

Table 8: Parámetros de salida generados por la rutina.

Parámetro	Tipo	Descripción	Unidad
mspace	double precision(26,*)	Matriz actualizada con propiedades físicas como posición, velocidad, densidad, presión y energía.	Variable (ver tabla de propiedades)
npairs	integer	Número máximo de interacciones de pares calculadas entre partículas.	-
nfilas	integer(*)	Cantidad de interacciones registradas por cada partícula fluida.	-
mrij, mviij	double precision	Matrices que almacenan distancias y diferencias de velocidades entre partículas.	-
mxij, mvxij	double precision	Matrices de diferencias espaciales y velocidades proyectadas para las interacciones de pares.	-

7.6 Rutinas `neighboring_search` y `neighborin_Grid_Hashing`

En esta etapa se identifican las partículas vecinas de cada partícula dentro de una longitud de suavizado (`hsm1`). Esta información es crucial para los cálculos posteriores, ya que los métodos SPH dependen de las interacciones entre partículas vecinas. Mediante el método directo rutina `neighboring_search` y mediante el método Grid Hashing rutina `neighborin_Grid_Hashing`.

Variables

Table 9: Parámetros de entrada y salida utilizados en la rutina `neighboring_Grid_Hashing`.

Variables	Tipo	Descripción	In / out
<code>rdomain</code>	<code>double precision</code>	Longitud de suavizamiento para determinar las interacciones entre partículas.	[in]
<code>mspace</code>	<code>array</code>	Arreglo multidimensional que almacena propiedades de las partículas, como coordenadas, velocidades, densidad, etc.	[in/out]
<code>ntype</code>	<code>integer</code>	Número de tipos de partículas en la simulación.	[in]
<code>npairs</code>	<code>integer</code>	Número máximo de pares de partículas que pueden interactuar.	[out]
<code>pairs</code>	<code>array</code>	Matriz que almacena los pares de partículas que interactúan.	[out]
<code>nfilas</code>	<code>array</code>	Número de interacciones por cada partícula del fluido.	[out]
<code>mrij</code>	<code>array</code>	Matriz que contiene las distancias entre partículas para cada interacción.	[out]
<code>mxij</code>	<code>array</code>	Matriz que almacena las diferencias de posiciones ($x_i - x_j$) entre partículas.	[out]
<code>mvij</code>	<code>array</code>	Matriz que contiene las diferencias de velocidades ($v_i - v_j$) entre partículas.	[in]
<code>mvxij</code>	<code>array</code>	Matriz que almacena las diferencias de velocidades en las componentes individuales ($v_{xi} - v_{xj}$).	[in]

Flujo de la rutina

La rutina de búsqueda directa realiza un procedimiento en el que compara las distancias entre todas las partículas, identificando aquellas que se encuentran dentro del rango de la longitud de suavizado. Las interacciones detectadas se almacenan en las matrices **npairs**, **nfilas**, **mrij**, **mvij**, **mxij** y **mvxij**, para ser utilizadas posteriormente en otras rutinas. Además, existe una versión paralelizada de esta rutina, que distribuye el cálculo de las distancias entre partículas en múltiples hilos de ejecución, mejorando significativamente la eficiencia del proceso.

Por otro lado, la rutina basada en el método de **Grid Hashing** utiliza una malla tridimensional que se ajusta al dominio de la simulación. Este método etiqueta todas las partículas, tanto las que representan el fluido como las que representan las fronteras, de acuerdo con la posición que ocupan dentro de la cuadrícula de la malla. Posteriormente, la búsqueda de partículas vecinas se limita únicamente a los cuadros adyacentes a la partícula de interés, lo que incrementa notablemente la eficiencia de la búsqueda. Al igual que en la rutina de búsqueda directa, todas las interacciones identificadas se registran en las mismas matrices mencionadas anteriormente.

7.7 Rutinas **wijdwij** y **kernel**

Esta rutina calcula la función de suavizado $W(r, h)$ y su derivada, donde r representa la distancia entre dos partículas y h es la longitud de suavizado.

Variables

Las variables de entrada de la rutina son, en su mayoría, las mismas que se utilizan en la rutina **neighboring_search**, además de los arreglos $W(r, h)$ y $\nabla_r W(r, h)$ son variables de salida, que almacenan los resultados del cálculo de la función de suavizado y su derivada para todas las interacciones entre partículas.

Flujo de la rutina

La rutina procesa todos los pares de interacciones y calcula la función de suavizado, almacenando los resultados en arreglos específicos. Este cálculo se realiza mediante la rutina interna **kernel**, que permite seleccionar entre diversas funciones de suavizado, como la Gaussiana, el Spline cúbico o el Spline quíntico. Además de evaluar la función de suavizado, **kernel** calcula sus derivadas espaciales, que son esenciales para los cálculos de gradientes en el método SPH.

7.8 Rutina **density**

Utilizando la función de suavizado, se calcula la densidad de cada partícula como:

$$\rho_i = \sum_j m_j W(r_{ij}, h_i),$$

donde m_j es la masa de la partícula j , $W(r_{ij}, h_i)$ es la función de suavizado evaluada en la distancia r_{ij} . Esto en la rutina **density**.

Variables

Las principales variables de la rutina incluyen la matriz **mspace**, que actúa como un arreglo de entrada y salida, y la variable **w**, que corresponde a la función de suavizado.

Flujo de la rutina

La rutina recibe como entrada la matriz **mspace** y modifica su componente **mspace(9,i)**, la cual representa la densidad. Esta modificación se realiza aplicando la ecuación previamente descrita, utilizando la función de suavizado **w**.

7.9 Rutina **presioni**

Esta rutina calcula la presión mediante la ecuación de estado:

$$p_i = B \left[\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right],$$

donde B , ρ_0 , y γ son parámetros específicos del modelo que se calculan en la rutina y el archivo de parámetros..

Variables

La principal variable de entrada y salida en esta rutina es la matriz **mspace**.

Flujo de la rutina

La rutina recibe como entrada la matriz **mspace** y modifica su componente **mspace(10,i)**, la cual representa la presión de cada partícula. Esta modificación se realiza aplicando la ecuación previamente descrita.

7.10 Rutina **momento**

La rutina **momento** calcula las fuerzas que actúan sobre las partículas de un sistema SPH, incluyendo fuerzas hidrodinámicas, viscosas y externas. Estas fuerzas son fundamentales para determinar la aceleración de cada partícula y resolver las ecuaciones de movimiento.

Variables

Las variables de la rutina son, en su mayoría, las mismas que se utilizan en la rutina **neighboring_search**, ver tabla 10. Principalmente, modifica los valores de la derivada de la velocidad en la variable **mspace(20 a 22)** mediante la ecuación de momento. Además, calcula la tasas de deformación, las fuerzas de viscosidad y fuerzas externas como la gravedad, ver tabla 11.

Parámetro	Tipo	Descripción	Unidad
<code>mpace</code>	<code>double precision(26,*)</code>	Matriz que contiene las propiedades físicas de las partículas, como posición, velocidad, presión, densidad, etc.	Variable
<code>ntype</code>	<code>integer(2)</code>	Número de partículas por tipo (fluidas, virtuales, etc.).	-
<code>npairs</code>	<code>integer</code>	Número máximo de interacciones entre partículas.	-
<code>pairs</code>	<code>integer(npairs, ntype(1))</code>	Matriz que relaciona las partículas con sus vecinos.	-
<code>nfilas</code>	<code>integer(ntype(1))</code>	Cantidad de interacciones vecinas registradas para cada partícula fluida.	-
<code>w</code>	<code>double precision(npairs, ntype(1))</code>	Función de suavizado evaluada para cada par de interacción.	-
<code>dwdx</code>	<code>double precision(3,*)</code>	Derivadas espaciales de la función de suavizado ($\partial W / \partial x, y, z$).	-
<code>mxij</code>	<code>double precision(3,*)</code>	Matriz de diferencias espaciales entre partículas vecinas.	-
<code>mvij</code>	<code>double precision(*)</code>	Matriz de diferencias de velocidad entre partículas vecinas.	-
<code>mvxij</code>	<code>double precision(3,*)</code>	Matriz de diferencias de velocidades proyectadas.	-

Table 10: Parámetros de entrada de la rutina `tasa_deformacion_epsilon`.

Flujo de la rutina

El flujo de la rutina se describe de la siguiente manera:

1. **Inicialización:** Se inicializan las aceleraciones de todas las partículas en `mpace(20:22, i)`. Posteriormente, se calculan parámetros iniciales como la longitud de suavizado y las matrices de interacción.
2. **Cálculo de Presión:** Se invoca la subrutina `sph_presion` para calcular la primera parte de la ecuación de momento, correspondiente a las fuerzas de presión, utilizando las densidades almacenadas en `mpace(9, i)` y la función de suavizado.
3. **Cálculo de la Tasa de Deformación:** Dependiendo del modelo de viscosidad

Parámetro	Tipo	Descripción	Unidad
<code>mspace(20:22, i)</code>	<code>double precision</code>	Aceleraciones calculadas en cada dirección (x, y, z) para cada partícula.	-
<code>epsilon</code>	<code>double precision(6,*)</code>	Tasa de deformación de las partículas (componentes tensoriales).	-
<code>viscforce</code>	<code>double precision(9,*)</code>	Fuerzas viscosas calculadas en función del modelo de viscosidad seleccionado.	-
<code>extforce</code>	<code>double precision(3,*)</code>	Fuerzas externas aplicadas a cada partícula en las direcciones (x, y, z).	-

Table 11: Parámetros de salida de la rutina `momento`.

seleccionado (`mvisc`), se utiliza la rutina `tasa_deformacion_epsilon` para calcular el tensor de deformación en el caso de viscosidad constante. Si en el archivo de parámetros se selecciona viscosidad no constante, el código utiliza la rutina `tasa_deformacion_epsilon2`, que implementa un modelo avanzado para evaluar la deformación y calcular un coeficiente de viscosidad dependiente de la misma, aplicable a modelos no-Newtonianos.

- Cálculo de Fuerzas Viscosas:** Si `mvisc == 1`, se invoca `viscous_force`, que calcula fuerzas viscosas básicas. En caso de que `mvisc == 2`, se utiliza `viscous_force_mud`, que implementa un modelo dependiente de la viscosidad no lineal.
- Cálculo de Fuerzas Externas:** Se llama a `external_force` para incorporar efectos como gravedad u otras fuerzas específicas del dominio.
- Actualización de Aceleraciones:** Finalmente, las aceleraciones resultantes se calculan combinando las contribuciones de fuerzas viscosas, externas y de presión. Estas se almacenan en `mspace(20:22, i)`.

En resumen, la rutina `momento` calcula las fuerzas resultantes para cada partícula, considerando presión, viscosidad y fuerzas externas. Estas fuerzas se utilizan para determinar las aceleraciones necesarias para resolver las ecuaciones de movimiento del sistema en el esquema SPH. La rutina organiza las interacciones en matrices que facilitan una evaluación eficiente y estructurada de todas las contribuciones a las fuerzas en el sistema.

7.11 Rutina `sph_presion`

La rutina `sph_presion` se utiliza en el método SPH (*Smoothed Particle Hydrodynamics*) para calcular el término de gradiente de presión que actúa sobre cada partícula del sistema.

Este término es una fuerza clave en las ecuaciones de movimiento, ya que controla el comportamiento hidrodinámico de las partículas en función de su presión y densidad.

Variables

Las variables de la rutina son, en su mayoría, las mismas que se utilizan en la rutina `momento`. Principalmente, modifica los valores de la derivada de la velocidad en la variable `mspace(20 a 22)` en la componente de presión de la ecuación de momento.

Flujo de la rutina

El propósito general es evaluar las fuerzas hidrodinámicas basadas en el gradiente de presión, utilizar la relación entre partículas vecinas para calcular las contribuciones de presión y densidad a través de una función de suavizado (*kernel*) y sus derivadas para actualizar los valores de aceleración debido a la presión en las partículas. El flujo general de la rutina se puede describir como:

1. **Inicialización:** se inicializan las aceleraciones de las partículas (`mspace(20:22, i)`) en cero. Luego se calcula $\frac{p_i}{\rho_i^2}$ para cada partícula i , donde p_i es la presión y ρ_i es la densidad.
2. **Cálculo del Gradiente de Presión:** para cada partícula i , se evalúan sus interacciones con las partículas vecinas j listadas en la matriz `pairs`. Entonces, se calcula el término combinado de presión y densidad para cada par i, j :

$$\text{mprhoij} = m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right),$$

donde m_j es la masa de la partícula vecina j , y se evalúan las contribuciones al gradiente de presión en cada dirección (x, y, z) :

$$\text{sumx} = \sum_j \text{mprhoij} \frac{\partial W}{\partial x}, \quad \text{sumy} = \sum_j \text{mprhoij} \frac{\partial W}{\partial y}, \quad \text{sumz} = \sum_j \text{mprhoij} \frac{\partial W}{\partial z}.$$

3. **Actualización de Aceleraciones:** se resta la suma acumulada en cada dirección para actualizar las aceleraciones de la partícula i :

$$\text{mspace}(20, i) = -\text{sumx}, \quad \text{mspace}(21, i) = -\text{sumy}, \quad \text{mspace}(22, i) = -\text{sumz}.$$

4. **Iteración sobre Todas las Partículas:** finalmente, se repite el proceso para todas las partículas del tipo definido en `ntype`, considerando solo las partículas fluidas.

En resumen la rutina `sph_presion` implementa el cálculo del gradiente de presión en el método SPH, evaluando las contribuciones de cada par de partículas vecinas mediante una función de suavizado y sus derivadas espaciales. El resultado es la aceleración en cada partícula debido a la presión, que se utiliza posteriormente para avanzar en el tiempo las posiciones y velocidades en el modelo SPH.

7.12 Rutina tasa_deformacion_epsilon

La subrutina `tasa_deformacion_epsilon` calcula el tensor de tasas de deformación (ε) para cada partícula en un sistema SPH. Este tensor describe los cambios de velocidad y deformación en las partículas debido a interacciones con sus vecinas, y es fundamental para modelar fenómenos como la viscosidad y la disipación energética en fluidos.

Variables

La mayoría de las variables utilizadas en esta rutina son las mismas que se emplean en la rutina `momento` (ver Tabla 10). Su principal función es actualizar los valores de la derivada de la velocidad en la variable `mpace(20:22)`, centrándose específicamente en la segunda componente de la ecuación de momento, correspondiente a la tasa de deformación (ver Tabla 12).

Parámetro	Tipo	Descripción	Unidad
<code>epsilon(1,i)</code>	double precision	Componente ε^{xx} del tensor de tasas de deformación.	-
<code>epsilon(2,i)</code>	double precision	Componente $\varepsilon^{xy} = \varepsilon^{yx}$ del tensor de tasas de deformación.	-
<code>epsilon(3,i)</code>	double precision	Componente $\varepsilon^{xz} = \varepsilon^{zx}$ del tensor de tasas de deformación.	-
<code>epsilon(4,i)</code>	double precision	Componente ε^{yy} del tensor de tasas de deformación.	-
<code>epsilon(5,i)</code>	double precision	Componente $\varepsilon^{yz} = \varepsilon^{zy}$ del tensor de tasas de deformación.	-
<code>epsilon(6,i)</code>	double precision	Componente ε^{zz} del tensor de tasas de deformación.	-

Table 12: Parámetros de salida de la rutina `tasa_deformacion_epsilon`.

Flujo de la rutina

El propósito general es evaluar las tasas de deformación tensorial para cada partícula utilizando las velocidades relativas y las derivadas espaciales de la función de suavizado (*kernel*). Calcula componentes diagonales y no diagonales del tensor, esenciales para modelos de flujo viscoso y no lineal. El flujo de la rutina se muestra a continuación:

1. **Inicialización:** las componentes del tensor de tasas de deformación (ε) se inicializan a cero y se calcula la contribución de densidades vecinas a través de la relación $\frac{m_j}{\rho_j}$.

2. **Cálculo de Componentes del Tensor:** se evalúan las contribuciones diagonales ($\varepsilon^{xx}, \varepsilon^{yy}, \varepsilon^{zz}$) mediante las diferencias de velocidad proyectadas y las derivadas del kernel:

$$\varepsilon^{xx} = 2 \cdot \text{sumexx} - \frac{2}{3} \cdot \text{sumdivg}, \quad \text{y así para } \varepsilon^{yy} \text{ y } \varepsilon^{zz}.$$

Las componentes no diagonales ($\varepsilon^{xy}, \varepsilon^{xz}, \varepsilon^{yz}$) se obtienen sumando contribuciones cruzadas:

$$\varepsilon^{xy} = \text{sumexy1} + \text{sumexy2}, \quad \text{y similar para las otras componentes.}$$

3. **División del Gradiente de Velocidades:** se calcula $\nabla \cdot \vec{v}$, utilizado para ajustar las contribuciones diagonales del tensor.
4. **Iteración para Todas las Partículas:** se repite el proceso para cada partícula fluida, considerando solo las vecinas listadas en **pairs**.

En resumen, la subrutina `tasa_deformacion_epsilon` calcula el tensor de tasas de deformación para cada partícula en un sistema SPH, utilizando las velocidades relativas y derivadas del kernel. Este tensor es esencial para modelar comportamientos viscosos y analizar la dinámica de fluidos complejos en simulaciones numéricas.

7.13 Rutina `tasa_deformacion_epsilon2`

Esta rutina realiza el mismo procedimiento que la rutina `tasa_deformacion_epsilon` pero adicional calcula el producto tensorial de la ecuación 30.

7.14 Rutina `viscous_force`

La subrutina `viscous_force` calcula las fuerzas viscosas que actúan entre las partículas de un sistema SPH (*Smoothed Particle Hydrodynamics*). Estas fuerzas dependen de las tasas de deformación (ε) previamente calculadas y se utilizan para modelar el comportamiento viscoso del fluido, particularmente en simulaciones con fluidos no newtonianos o multifase (ver tabla 13).

Variables

La mayoría de las variables empleadas en esta rutina coinciden con las utilizadas en la rutina `momento`. En particular, la matriz `viscforce(9,i)` se utiliza para calcular las fuerzas de viscosidad, correspondientes a la segunda parte de la ecuación de momento

Flujo de la rutina

El propósito general es calcular las componentes del tensor de fuerzas viscosas (F_{viscosa}) entre partículas vecinas. Además, incorpora la viscosidad dinámica y la tasa de deformación en las interacciones. Evalúa las contribuciones de los pares de partículas vecinas utilizando las derivadas espaciales del kernel y las propiedades de densidad y masa de las partículas. El flujo general de la rutina se describe a continuación:

Parámetro	Tipo	Descripción	Unidad
viscforce(9,i)	double precision	Componentes de las fuerzas viscosas en todas las direcciones: $F^{xx}, F^{xy}, F^{xz}, \dots, F^{zz}$.	-

Table 13: Parámetros de salida de la rutina `viscous_force`.

- Inicialización de Variables:** se inicializan las componentes de las fuerzas viscosas (`sumvisxx`, `sumvisxy`, ...) a cero, se calcula la densidad al cuadrado (ρ_i^2) y se divide cada componente de las tasas de deformación por este valor para obtener $\frac{\varepsilon}{\rho_i^2}$.
- Cálculo de Contribuciones Viscosas:** para cada partícula i , se itera sobre todas sus partículas vecinas j listadas en la matriz `pairs`, luego se evalúan las contribuciones a las fuerzas viscosas en cada dirección utilizando la fórmula general:

$$F^{xx} = \sum_j m_j \left(\frac{\varepsilon_i^{xx}}{\rho_i^2} + \frac{\varepsilon_j^{xx}}{\rho_j^2} \right) \frac{\partial W}{\partial x},$$

donde W es el kernel de suavizado, y ε_i^{xx} y ε_j^{xx} son las tasas de deformación en la dirección xx para las partículas i y j , respectivamente.

- Actualización de las Fuerzas Viscosas:** las contribuciones acumuladas (`sumvisxx`, `sumvisxy`, ...) se multiplican por la viscosidad dinámica (μ) para calcular las fuerzas finales:

$$\text{viscforce}(1, i) = \mu \cdot \text{sumvisxx}, \quad \text{viscforce}(2, i) = \mu \cdot \text{sumvisxy}, \quad \text{etc..}$$

- Iteración sobre Todas las Partículas:** el proceso se repite para todas las partículas del sistema, considerando solo las fluidas y sus vecinas.

En resumen, la subrutina `viscous_force` implementa el cálculo de las fuerzas viscosas en un sistema SPH, considerando las tasas de deformación y las propiedades del fluido como la densidad y la viscosidad dinámica. Los resultados son fundamentales para modelar la interacción entre partículas en simulaciones de fluidos viscosos, proporcionando una representación precisa de la disipación de energía en el sistema.

7.15 Rutina `viscous_force_mud`

Esta rutina realiza cálculos similares a los de la rutina `viscous_force`, pero introduce una importante diferencia: en lugar de asumir una viscosidad constante para todo el sistema, utiliza la viscosidad específica de cada partícula. En la segunda parte de la ecuación de momento, la viscosidad de cada partícula es incluida de manera explícita, lo que permite modelar fluidos multifase con mayor precisión.

La formulación general se expresa como:

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\alpha} + \sum_{j=1}^N m_j \left(\frac{\eta_i \varepsilon_i^{\alpha\beta}}{\rho_i^2} + \frac{\eta_j \varepsilon_j^{\alpha\beta}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} + g^\alpha, \quad (36)$$

En esta ecuación, la primera suma corresponde a la contribución de las fuerzas de presión, mientras que la segunda incorpora las fuerzas viscosas dependientes de las tasas de deformación $\varepsilon^{\alpha\beta}$ y la viscosidad dinámica específica η de cada partícula. El término g^α representa la aceleración gravitacional en la dirección correspondiente.

7.16 Rutina `viscous_model`

Esta rutina implementa un modelo de viscosidad específico para cada partícula, basado en el modelo de Herschel-Bulkley, descrito en la ecuación 35. Este modelo captura el comportamiento reológico del fluido, calculando el factor de viscosidad a partir del segundo invariante del tensor de la tasa de deformación.

7.17 Rutina `external_force`

Esta rutina se encarga de añadir las fuerzas externas que afectan el movimiento de las partículas. En el caso de simulaciones terrestres, implementa específicamente la fuerza de gravedad actuando sobre la componente z . De este modo, la aceleración calculada a partir de la ecuación de momento se complementa sumando este término adicional.

7.18 Archivo de Parámetros del Código SPH, `param.inc`.

Este archivo define todos los parámetros y constantes necesarias para el funcionamiento del código SPH. A continuación, se detallan cada uno de los parámetros junto con su descripción y valores predeterminados:

- **Constante matemática π :**
 - **pi:** Representa el valor de π , utilizado en múltiples cálculos.
 - **Valor:** 3.14159265358979323846.
- **Dimensión del problema:**
 - **dim:** Define la dimensionalidad del problema, permitiendo realizar simulaciones en 1D, 2D o 3D.
 - **Valor:** 3 (simulación en 3D).
- **Número máximo de partículas:**
 - **nmax:** Define la capacidad máxima de partículas en la simulación.
- **Dominio de simulación:**

- **dxmin, dxmax:** Límite inferior y superior en el eje x .
- **dymin, dymax:** Límite inferior y superior en el eje y .
- **dzmin, dzmax:** Límite inferior y superior en el eje z .
- **Valores:**

- **Método de búsqueda de vecinos:**

- **cns:** Selecciona el método de búsqueda para identificar las partículas vecinas.
 - * 1: Búsqueda directa.
 - * 2: Búsqueda paralelizada directa.
 - * 3: Grid Hashing.
 - * 4: Grid Hashing paralelizada.
- **Valor:** 3 (Grid Hashing).
- **np:** Número de procesadores a utilizar si **cns** = 4.

- **Configuración temporal:**

- **initime, fintime:** Define el paso inicial y final de la simulación (en número de pasos, no en segundos).
- **dt0:** Paso de tiempo para la integración.
- **Valores:**

- **Dominio de soporte:**

- **h0:** Longitud inicial de suavizado, depende de la distancia inicial entre las partículas.
- **kappa0:** Constante del dominio de soporte $kappa0 = 2.0$.

- **Función de suavizado:**

- **skf:** Define la función de suavizado a utilizar.
 - * 1: Spline cúbico (Monaghan 1985).
 - * 2: Gaussiana (Gingold y Monaghan 1981).
 - * 3: Spline quíntico (Morris 1997).

- **Altura máxima del bloque:**

- **ht:** Define la altura máxima del bloque en metros. Esta altura se establece dependiendo de la altura máxima del fluido en relación al punto más bajo de la superficie frontera.

- **Gravedad:**

- **g:** Aceleración gravitacional en z .

- **Valor:** 9.82.
- **Ecuación de estado artificial:**
 - **gamma:** Exponente adiabático.
 - **beta:** Factor de amortiguación.
 - **rho0:** Densidad inicial de las partículas.
 - **Valores:** $\gamma = 7.0$, $\beta = 1.0$, $\rho_0 = 1000.0$.
- **Viscosidad:**
 - **mu:** Coeficiente de viscosidad dinámica.
 - **muc:** Factor de ajuste de viscosidad.
 - **mvisc:** Modelo de viscosidad (1 = constante, 2 = variable).
 - **nn:** Índice de comportamiento reológico.
 - **Valores:** $\mu = 1.002 \times 10^{-3}$, $muc = 1000.0$, $mvisc = 1$, $nn = 1.5$.
- **Fuente de partículas:**
 - **source_mode:** Activa el modo fuente ('y' para activarlo).
 - **QQ:** Flujo de la fuente en $[m^2/s]$.
 - **start_x, start_y, start_z:** Coordenadas iniciales de la fuente.
 - **angle:** Ángulo de inclinación de la fuente.
 - **v_fuente0:** Velocidad inicial de la fuente.
 - **d_horizontal:** Distancia horizontal de la fuente.
 - **Valores:**
- **Frecuencia de guardado:**
 - **save_step:** Número de pasos entre cada guardado de resultados. **Valor:** 40 pasos.

Parámetro	Descripción y Valor
dim	Dimensión del problema (1D, 2D o 3D). Valor: 3 (simulación en 3D).
nmax	Número máximo de partículas permitidas. Valor: 60,000.
Dominio de Simulación dxmin, dxmax dymin, dymax dzmin, dzmax	Define los límites del dominio en cada eje: Límite en x . Valor: [948,400.0, 948,800.0]. Límite en y . Valor: [590,600.0, 591,000.0]. Límite en z . Valor: [2300.0, 2400.0].
cns	Método de búsqueda de vecinos: 1 = búsqueda directa, 2 = búsqueda paralelizada directa, 3 = Grid Hashing, 4 = Grid Hashing paralelizada, 5 = Grid Hashing implementa medida de distancia para gradiente de concentración en modelos de lodos. Valor: 3.
np	Número de procesadores si $cns = 4$. Valor: 8.
Pasos de Tiempo initime, fintime inioutfile dt0	Tiempo inicial y final en pasos (no en segundos). Valor: [0, 50,000]. Archivo inicial de salida. Valor: 'snapshot_000'. Paso de tiempo para la integración. Valor: 0.005.
Dominio de Soporte h0, kappa0	Longitud de suavizado inicial y constante del dominio. Valor: [1.0, 2.0].
skf	Función de suavizado: 1 = spline cúbico, 2 = Gaussiana, 3 = spline quintico. Valor: 1.
ht	Altura máxima del bloque en metros. Valor: 1.0.
g	Aceleración de la gravedad en z . Valor: 9.82.
Ecuación de Estado Artificial gamma, beta, rho0	Constantes para la densidad inicial. Valor: [7.0, 1.0, 1000.0].
Viscosidad mu, muc	Coefficientes de viscosidad dinámica. Valor: [1.002e-3, 1000.0].
Modelos Viscosidad mvisc mvisk nn	Modelo de viscosidad: 1 = constante (Fluido Newtoniano), 2 = viscosidad variable (Fluido No-Newtoniano, modelo de Herschel-Bulkey). Valor: 1. Modelo de viscosidad: 1 = parametro de viscosidad constante constante (modelo de Herschel-Bulkey K =constante), 2 = parametro de viscosidad variable (modelo de Herschel-Bulkey K =depende de la concentración). Valor: 1. Índice de comportamiento reológico. Valor: 1.5.
Fuente de Partículas source_mode QQ, mu_new, length start_x, start_y, start_z, angle v_fuente0, d_horizontal save_step	Activa el modo de fuente de partículas ('y'). Flujo, viscosidad de la fuente y longitud de la fuente. Valor: [11.5, 1.002e-3, 0.1]. Posición inicial y ángulo de la fuente. Valor: [948,452.3, 590,680.8, 2348, 90.0]. Velocidad de la fuente y distancia horizontal. Valor: [1.0, 17.0]. Frecuencia para guardar resultados. Valor: 40 pasos.

Table 14: Descripción del archivo de parámetros para el código SPH.

Referencias

- Altomare, C., Crespo, A. J., Rogers, B., Dominguez, J., Gironella, X., & Gómez-Gesteira, M. 2014, *Computers & Structures*, 130, 34
- Antuono, M., Colagrossi, A., Marrone, S., & Lugni, C. 2011, *Computer Physics Communications*, 182, 866
- Bertevas, E., Férec, J., Khoo, B. C., Ausias, G., & Phan-Thien, N. 2018, *Physics of Fluids*, 30, 103103
- bin Liu, M., ru Shao, J., & qi Li, H. 2013, *Journal of Hydrodynamics*, 25, 673
- Cáceres, J. A. 2011, Universidad Distrital Francisco José de Caldas
- Cao, Z., Patra, A., Bursik, M., Pitman, E. B., & Jones, M. 2018, *Geoscientific Model Development*, 11, 2691
- Capone, T., Panizzo, A., & Monaghan, J. J. 2010, *Journal of Hydraulic Research*, 48, 80
- Carrillo, E., Contreras, N., Sánchez, F., Alpala, J., Pulgarín, B., Medina, E., Reyes, A., Gamboa, C., Muñoz, C., Galarza, J., Laverde, C., & Agudelo, A. 2023, Informe visita de emergencia a la microcuenca de la quebrada Chontaduro y concepto técnico sobre el trazado alterno de la vía Panamericana – Municipio de Rosas (SGC)
- Chaparro-Cordón, J. L., Rodríguez-Castiblanco, É. A., Rangel-Flórez, M. S., García-Delgado, H., & Medina-Bello, E. 2020, SCG-XIII INTERNATIONAL SYMPOSIUM ON LANDSLIDES
- Crespo, A., Gómez-Gesteira, M., & Dalrymple, R. 2007, *Journal of Hydraulic Research*, 45, 631
- Cuomo, G., Panizzo, A., & Dalrymple, R. 2007, in *Coastal Engineering 2006: (In 5 Volumes)* (World Scientific), 274–286
- Dalrymple, R. A., & Rogers, B. 2006, *Coastal engineering*, 53, 141
- De Leffe, M., Le Touzé, D., & Alessandrini, B. 2010, *Journal of Hydraulic Research*, 48, 118
- Del Guzzo, A., & Panizzo, A. 2007, *sphe*, 161
- Domínguez, J., Crespo, A., Gómez-Gesteira, M., & Marongiu, J. 2011, *International Journal for Numerical Methods in Fluids*, 67, 2026
- Duan, G., Chen, B., Zhang, X., & Wang, Y. 2017, *Computer Methods in Applied Mechanics and Engineering*, 320, 133

- Ellero, M., Kröger, M., & Hess, S. 2002, *Journal of Non-Newtonian Fluid Mechanics*, 105, 35
- Enríquez, J. E. I. 2012, *Revista Unimar*, 30
- Fan, X.-J., Tanner, R., & Zheng, R. 2010, *Journal of Non-Newtonian Fluid Mechanics*, 165, 219
- Fourtakas, G. 2014, PhD thesis, University of Manchester
- Fourtakas, G., Rogers, B. D., & Laurence, D. R. 2013, *La Houille Blanche*, 39
- Fraga Filho, C. A. D., Fraga Filho, C. A. D., & Castro. 2019, *Smoothed Particle Hydrodynamics* (Springer)
- Gallo, A. M. G., Murcia, H. F., Jiménez, G. P. C., & Laverde, C. A. 2011, *Geología Colombiana*, 36, 135
- Gingold, R. A., & Monaghan, J. J. 1977, *Monthly notices of the royal astronomical society*, 181, 375
- Gómez-Gesteira, M., Cerqueiro, D., Crespo, C., & Dalrymple, R. 2005, *Ocean Engineering*, 32, 223
- Gomez-Gesteira, M., & Dalrymple, R. 2004, *Ocean Engineering*, ASCE, 130, 63
- Goswami, P., Schlegel, P., Solenthaler, B., & Pajarola, R. 2010
- Gross, J., Köster, M., & Krüger, A. 2019, in *CGVC*, 55–63
- Haddad, B., Palacios, D., Pastor, M., & Zamorano, J. J. 2016, *Journal of Volcanology and Geothermal Research*, 324, 73
- Hashemi, M., Fatehi, R., & Manzari, M. 2011, *Journal of Non-Newtonian Fluid Mechanics*, 166, 1239
- Laigle, D., Lachamp, P., & Naaim, M. 2007, *Computational Geosciences*, 11, 297
- Lee, E.-S., Violeau, D., Issa, R., & Ploix, S. 2010, *Journal of Hydraulic Research*, 48, 50
- Liu, M., & Liu, G. 2010, *Archives of computational methods in engineering*, 17, 25
- López, D., Marivela, R., & Garrote, L. 2010, *Journal of Hydraulic Research*, 48, 142
- Lucy, L. B. 1977, *AJ*, 82, 1013
- Manenti, S., Sibilla, S., Gallati, M., Agate, G., & Guandalini, R. 2011, in *Proceedings of the III International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2011)*, Corfu, Greece, 25–28

- Manenti, S., Sibilla, S., Gallati, M., Agate, G., & Guandalini, R. 2012, *Journal of Hydraulic Engineering*, 138, 272
- Marongiu, J.-C., Leboeuf, F., Caro, J., & Parkinson, E. 2010, *Journal of Hydraulic Research*, 48, 40
- Mokos, A. D. 2014, The University of Manchester
- Murcia, H. F., & Macías, J. L. 2009, *Revista mexicana de ciencias geológicas*, 26, 1
- Panizzo, A., Cuomo, G., & Dalrymple, R. A. 2007, in *Coastal Engineering 2006: (In 5 Volumes)* (World Scientific), 1503–1515
- Panizzo, A., & Dalrymple, R. 2005, in *Coastal Engineering 2004: (In 4 Volumes)* (World Scientific), 1147–1159
- Polwaththe-Gallage, H.-N., Saha, S. C., Sauret, E., Flower, R., & Gu, Y. 2016a, *International Journal of Mechanics and Materials in Design*, 12, 477
- Polwaththe-Gallage, H.-N., Saha, S. C., Sauret, E., Flower, R., Senadeera, W., & Gu, Y. 2016b, *Biomedical engineering online*, 15, 161
- Pries, A. R., Secomb, T. W., Gaehtgens, P., & Gross, J. 1990, *Circulation research*, 67, 826
- Rajasekaran, S., & Reif, J. 2007, *Handbook of parallel computing: models, algorithms and applications* (CRC press)
- Rogers, B. D., & Dalrymple, R. A. 2008, in *Advanced numerical models for simulating tsunami waves and runup* (World Scientific), 75–100
- Santamaría, J. E. V., Vélez, M. I. G., & Hincapié, H. D. M. 2018, *Revista de derecho*, 145
- Schmid-Schönbein, G., Skalak, R., Usami, S., & Chien, S. 1980, *Microvascular research*, 19, 18
- Scott, K. M., Vallance, J. W., Kerle, N., Luis Macías, J., Strauch, W., & Devoli, G. 2005, *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group*, 30, 59
- Shao, J., Li, H., Liu, G., & Liu, M. 2012, *Computers & Structures*, 100-101, 18
- Shao, S., & Gotoh, H. 2004, *Coastal Engineering Journal*, 46, 171
- Springel, V. 2005, *MNRAS*, 364, 1105
- Vacondio, R., Aureli, F., Ferrari, A., Mignosa, P., & Dal Palù, A. 2016, *Natural Hazards*, 80, 103
- Vacondio, R., Mignosa, P., Rogers, B., & Stansby, P. 2010, *River Flow 2010*, 1621

- Vakilha, M., & Manzari, M. T. 2008, Transport in porous media, 74, 331
- Viccone, G., Bovolín, V., & Carratelli, E. P. 2008, International Journal for Numerical Methods in Fluids, 58, 625
- Viero, D. P., & Valipour, M. 2017, Advances in Water Resources, 104, 1
- Wang, W., Chen, G., Han, Z., Zhou, S., Zhang, H., & Jing, P. 2016, Natural Hazards, 81, 1981
- Xu, X. & Ouyang, J. 2013, Journal of Non-Newtonian Fluid Mechanics, 202, 54
- Yang, X., & Kong, S.-C. 2017, Physical Review E, 96, 033309
- Ye, T., Pan, D., Huang, C., & Liu, M. 2019, Physics of Fluids, 31, 011301
- Zhang, D., Shi, Y., Huang, C., Si, Y., Huang, B., & Li, W. 2018, Ocean Engineering, 152, 273
- Zhang, Y., Ban, X., Guo, Y., & Obaidat, M. S. 2020, Simulation Modelling Practice and Theory, 102212
- Zhang, Z., & Liu, M. 2017, Engineering Analysis with Boundary Elements, 83, 141
- Zou, S., & Dalrymple, R. A. 2007, in Coastal Engineering 2006: (In 5 Volumes) (World Scientific), 2477–2484

Elaborado por:



Diego Valencia Enríquez

Fecha de actualización: 25 de Febrero de 2025

Documento creado en LaTeX mediante la plataforma Overleaf. LaTeX es un sistema de composición de documentos que se utiliza principalmente para la creación de documentos científicos y técnicos, como artículos académicos, libros, informes y presentaciones. Utiliza un lenguaje de marcado para estructurar y formatear el texto, lo que permite a los usuarios concentrarse en el contenido del documento sin preocuparse por el diseño visual. LaTeX es especialmente popular en la comunidad académica debido a su capacidad para producir documentos con una apariencia profesional y su soporte para la escritura matemática compleja y la generación automática de referencias bibliográficas.