

**MODELO DE SIMULACIÓN Y VISUALIZACIÓN DEL PROCESO DE
MEZCLADO ULTRASÓNICO PARA LA PRODUCCIÓN DE BIODIESEL**

por

Luis Fernando De La Torre Quintana

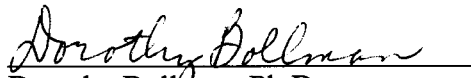
Tesis sometida en cumplimiento parcial
de los requisitos para el grado de

MAESTRO EN CIENCIAS
en
Computación Científica

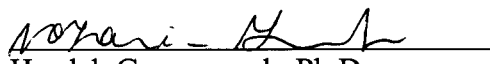
**UNIVERSIDAD DE PUERTO RICO
RECINTO UNIVERSITARIO DE MAYAGÜEZ**

2004

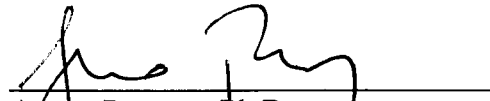
Aprobado por:


Dorothy Bollman, Ph.D.
Miembro, Comité Graduado

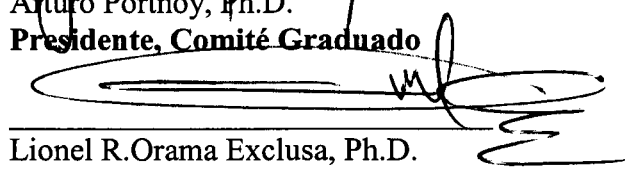
10/julio/04
Fecha


Haedeh Gooransarab, Ph.D.
Miembro, Comité Graduado

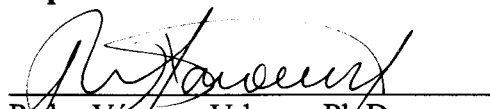
12/julio/2004
Fecha


Arturo Portnoy, Ph.D.
Presidente, Comité Graduado

12/julio/2004
Fecha


Lionel R. Orama Exclusa, Ph.D.
Representante de Estudios Graduados

9/julio/04
Fecha


Pedro Vázquez Urbano, Ph.D.
Director de Departamento

7/12/04
Fecha

Abstract

Biodiesel is an “environmental” and renewable alternative to fuel diesel engines. In the production process of Biodiesel, it is necessary to mix immiscible oils and alcohols. This production process needs fast, effective and viable methods. Thus, in the search to optimize the process an alternative mixing method has been developed by means of ultrasound, which promises to be faster, and more efficient and viable than traditional methods. This method generates cavities or bubbles that implode generating local "jets" that mix. Starting for the study of a single bubble that receives an ultrasonic stimulation, we will be able to find an approximation of the energy released by it and its lifetime, and thus have a model for the bubble. The model that we propose is based on the lattice in \mathbb{R}^3 space, in which each point is used as reference for the possible appearance of a cavity and depending on the location and the influence of the other bubbles a possible variation in the ultrasonic stimulate that it will receive is calculated. Then the model for a single bubble is used to obtain the lifetime and the energy emitted by the cavity. This will give the moment and the amount of mass that needs to be moved around this. We develop a visual model that allows us to see with clarity which happens during the mixing process. It is hoped that the model obtained will reflect the essence of the process, and will give a virtual laboratory where simulations can be made and design decisions can be taken at a very low cost. This promises to be a very attractive alternative for reactor design.

Resumen

El biodiesel es una alternativa "ambiental" y renovable para alimentar motores diesel. En el proceso de producción de biodiesel es necesario mezclar aceites y alcoholes (líquidos inmiscibles). Este proceso de producción necesita de métodos rápidos, efectivos y económicos. Así, en la búsqueda de optimizar el proceso, se ha desarrollado un método alternativo de mezclado por medio de ultrasonido, el cual promete ser más rápido, eficiente y económico que métodos tradicionales. Este método genera cavidades o burbujas que implotan generando "jets" (chorros) locales que mezclan. Tomando como base el estudio de una burbuja que recibe un estímulo ultrasónico, podremos encontrar una aproximación de la energía liberada por esta y su tiempo de vida, y así tener un modelo para una burbuja. El modelo que proponemos se basa en un mallado en el espacio \mathbb{R}^3 , en el que cada punto es usado como referencia para la posible aparición de una cavidad y dependiendo de la localización y de la influencia de las demás burbujas se calcula una posible variación en el estímulo ultrasónico que esta recibirá, Entonces se usa el modelo de una burbuja para obtener el tiempo de vida y la energía emitida por la cavidad, esto nos dará el momento y la cantidad de masa que moverá el "jets" alrededor de esta. Desarrollaremos un modelo visual que nos permita ver con claridad lo que sucede durante el proceso de mezclado. De obtener un modelo que refleje la esencia del proceso, tendremos un laboratorio virtual en donde se pueden realizar simulaciones y hacer pruebas de diseño a un costo muy bajo. Esta promete ser una alternativa de diseño muy atractiva.

*A mi madre Maria Cristina
y a mi amor Oliva*

AGRADECIMIENTOS

Quiero dar gracias a Dios por la vida, la salud. Sin su ayuda esta oportunidad no hubiera sido posible.

Al Departamento de Matemáticas de la Universidad de Puerto Rico, Recinto de Mayagüez, por las facilidades brindadas para la realización de este trabajo.

A mi director de tesis Dr Arturo Portnoy por ser mi guía durante todo el desarrollo del trabajo.

A Dr. Pedro Vasquez, Director del Departamento de Matemáticas por sus regaños y sus buenos consejos.

A Carlos Beltrán por el trabajo realizado sobre el estudio de una Burbuja.

A Eliseo y Leonardo por sus buenos consejos.

Por ultimo a toda familia y en especial a mi madre y a mi novia por su apoyo incondicional.

Tabla de Contenido

| | |
|---|----|
| 1. INTRODUCCIÓN | ii |
| 1.2 Justificación | 2 |
| 1.3 Objetivos | 3 |
| 1.4 Revisión de la literatura | 4 |
| 1.4.1 Modelos de fluidos..... | 4 |
| 1.4.2 Biodiesel | 5 |
| 2. ANÁLISIS DEL PROBLEMA..... | 7 |
| 2.1 Método de producción | 7 |
| 2.2 Uso de ultrasonido | 8 |
| 2.2.1 Por que mezclado ultrasónico | 8 |
| 2.2.2 Condiciones y parámetros necesarios | 9 |
| 2.3 Modelo general | 9 |
| 2.3.1 Cavitación, Implosión y Ultrasonidos..... | 12 |
| 2.3.2 Ecuacion de Rayleigh-Plesset | 13 |
| 2.3.3 Cavidades (semillas)..... | 14 |
| 2.3.4 Variación del ultrasonido..... | 14 |
| 2.3.5 Intercambio de masa..... | 15 |
| 3. EL MODELO..... | 16 |
| 3.1 Modelo estático..... | 16 |
| 3.2 Suposiciones del modelo..... | 18 |
| 3.3 Construcción del modelo | 19 |
| 3.3.1 Distribución (S_2)..... | 19 |
| 3.3.2 Conjunto C_x | 20 |
| 3.3.3 Factor \bar{R}_x | 22 |
| 3.3.4 Uso de la Función F (SI) | 22 |
| 3.4 Volumen..... | 24 |
| 3.4.1 Mallado del volumen | 24 |
| 3.4.2 Mezclado..... | 25 |

| | |
|---|----|
| 3.4.3 Cálculo de la masa | 25 |
| 3.4.4 Partículas..... | 26 |
| 3.5 Descripción del algoritmo..... | 27 |
| 3.5.1 Esquema general | 27 |
| 3.5.2 El algoritmo secuencial..... | 29 |
| 3.5.3 Complejidad del Algoritmo | 34 |
| 3.5.4 El uso del Paralelismo..... | 35 |
| 3.6 Método de Paralelización..... | 36 |
| 3.6.1 Implementación del Algoritmo | 38 |
| 3.6.2 El algoritmo con MPI..... | 38 |
| 3.6.3 La implementación..... | 39 |
| 3.7 Visualización..... | 41 |
| 3.7.1 Cavidades..... | 41 |
| 3.7.2 Mezcla (dos fases)..... | 43 |
| 3.7.3 Un valor ó razón que sirva como medida del mezclado | 43 |
| 4. RESULTADOS | 46 |
| 4.1 Implementación en Matlab | 46 |
| 4.1.1 Dos simulaciones | 46 |
| 4.2 Rendimiento del Programa | 49 |
| 4.2.1 Speedup..... | 49 |
| 4.3 Implementación C, MPI y Matlab. | 52 |
| 5. CONCLUSIONES | 55 |
| 6 TRABAJOS FUTUROS | 57 |
| BIBILOGRAFÍA | 58 |
| APENDICE 1..... | 59 |
| APENDICE 2..... | 59 |

Lista de Tablas

| | |
|---|----|
| Tabla 1. Propiedades del Biodiesel. | 5 |
| Tabla 2. Tiempos de computo algoritmo en C y algoritmo en Matlab | 36 |
| Tabla 3 .Constantes físicas..... | 46 |
| Tabla 4. Constantes para el modelo (primera simulación). | 47 |
| Tabla 5. Constantes para el modelo (segunda simulación)..... | 48 |
| Table 6. Contiene el “ <i>speedup</i> ” para el cálculo de la parte “semillas”..... | 50 |
| Table 7. Resultados de ejecutar el programa con una matriz A de 20^3 | 51 |
| Tabla 8. Resultados (160 iteraciones en una matriz A de 103) | 52 |
| Tabla 9. Parámetros físicos:..... | 60 |

Lista de Figuras

| | |
|---|----|
| Figura 1. Geometría del mallado. | 16 |
| Figura 2. Geometría del sistema. | 17 |
| Figura 3. Conjunto C_x | 21 |
| Figura 4. Intercambio de seis partículas. | 26 |
| Figure 5. El algoritmo secuencial. | 30 |
| Figure 6. Instantes de tiempo vs razón numérica que representa el mezclado. | 36 |
| Figura 7. Esquema de la técnica usada | 37 |
| Figure 8. Esquema de la técnica usada | 40 |
| Figura 9. Gráfica de la visualización de las cavidades. (mallado de 6^3)..... | 42 |
| Figura 10. (a) Dos fases (b) Superficie (c) Unión de a y b..... | 43 |
| Figura 11. Partículas distribuidas en un mezclado homogéneo..... | 44 |
| Figura 12. Función que encuentra una razón que mida el mezclado. | 45 |
| Figura 13. Dos tomas del mezclado (1) (a) iteracion 80. (b) iteracion 160. | 47 |
| Figura 14. Instantes de tiempo vs razón numérica que representa el mezclado. | 47 |
| Figura 15. Dos tomas del mezclado (a) en la iteración 80 y (b) en la iteración 160. | 48 |
| Figura 16. Instantes de tiempo vs razón numérica que representa el mezclado. | 49 |
| Figura 17. Gráfica del “ <i>speedup</i> ” Vs. aceleración lineal (semillas). | 50 |
| Figura 18. Gráfica del “ <i>speedup</i> ” Vs. aceleración lineal. (50 iteraciones con $N^3=20^3$) .. | 51 |
| Figura 19. Gráfica del “ <i>speedup</i> ” Vs. aceleración lineal. (160 iteraciones con $N^3=10^3$). 53 | 53 |
| Figura 20. Simulación en la iteración 80, usando un mallado de 10^3 cajas y un volumen de 30^3 partículas..... | 54 |
| Figura 21. Simulación en la iteración 160, usando un mallado de 10^3 cajas y un volumen de 30^3 partículas..... | 54 |
| Figure 22. El radio de una burbuja, R | 61 |

1. INTRODUCCIÓN

La cavitación ultrasónica (sección 2.3.1) es el fenómeno mediante el cual es posible entender el principio del mezclado por ultrasonido. Este método está siendo estudiado y aplicado en la producción de Biodiesel [5]. El ultrasonido ha probado ser una herramienta de gran utilidad en los procesos tales como el lavado quirúrgico, soldadura de plástico, pulido de moldes y otras muchas aplicaciones [7]. La aplicación de este en química orgánica y en particular en el proceso de producción de Biodiesel está en sus inicios [4] y la optimización de este proceso es una de las tareas a realizarse.

Para modelar este proceso sería necesario recurrir a conceptos de mecánica de fluido y lo primero que se viene a la mente es trabajar con las aproximaciones basadas en la solución numérica de las ecuaciones Navier-Stokes (modelos globales) y las aproximaciones microscópicas basadas en dinámica molecular (modelos locales).

Nosotros presentamos la idea de modelar este problema basándonos en alternativas que usan modelos locales simples y simulaciones para salvar limitaciones que se tienen para entender los modelos determinísticos clásicos (Ecuaciones de Navier Stokes); se partirá del modelo local hacia el comportamiento global con simulaciones construidas con heurísticas basadas en observaciones experimentales.

En este trabajo nosotros desarrollamos un modelo basado en el comportamiento de *una* burbuja dentro del proceso de producción de Biodiesel mediante mezclado ultrasónico. El sistema sobre el que trabajamos no consta de una sola burbuja, por tanto, usamos

modelos simplificados construidos con “heurísticas” basadas en observaciones experimentales y en principios fundamentales, que nos permiten extrapolar del estudio de una sola burbuja para tratar de predecir el comportamiento de muchas de ellas en el mismo sistema.

1.2 Justificación

El Biodiesel es una alternativa “ambiental” y renovable para alimentar motores diesel. En el proceso de producción de Biodiesel es necesario mezclar aceites y alcoholes, “líquidos inmiscibles”. Este proceso de producción necesita de métodos rápidos, efectivos y económicos. El proceso tradicional involucra mezclado mecánico y este es ineficiente y costoso. Así, en la búsqueda de optimizar el proceso, se ha desarrollado un método alternativo de mezclado por medio de ultrasonido, el cual promete ser más rápido, eficiente y económico comparado con métodos tradicionales. La viabilidad de este método está siendo estudiada y nuestros esfuerzos están encaminados a su optimización.

En este trabajo nos referiremos entonces al proceso de obtener Biodiesel mediante mezclado ultrasónico. Este método genera cavidades o burbujas que implotan generando “chorros” locales que mezclan. Basados en este concepto desarrollamos un modelo visual que nos permite ver con claridad lo que sucede en la esencia del proceso de mezclado y obtener un laboratorio virtual en donde se pueden realizar simulaciones y hacer pruebas de diseño a un costo muy bajo. Esta promete ser una alternativa de diseño muy atractiva.

1.3 Objetivos

- Desarrollar un modelo matemático que permita simular el mezclado de líquidos inmiscibles mediante la generación e implosión de cavidades por estimulación ultrasónica.
- Utilizar el MATLAB para generar una aplicación que nos permita tener una idea clara de cada una de las partes que componen el modelo que se está utilizando.
- Implementar los métodos numéricos necesarios y los algoritmos de simulación y visualización de forma eficiente incluyendo procesamiento en paralelo utilizando el estándar MPI (Message Passing Interface) [9].
- Desarrollar una interfase gráfica y una estrategia de visualización apropiada para visualizar los resultados numéricos de las simulaciones.
- Un objetivo inmediato de las simulaciones iniciales es demostrar que los “chorros” locales inducidos por las implosiones de cavidades son suficientes para generar un mezclado o emulsión muy fina, sin la aparición de flujos globales asociados al mezclado mecánico tradicional.

Este trabajo es parte de un proyecto extendido. Uno de los objetivos de este proyecto es utilizar simulaciones para hacer mejoras en el diseño de reactores. Este trabajo es el primer paso en esa dirección.

1.4 Revisión de la literatura

Esta sección trata sobre la utilización de observaciones experimentales como heurísticas para un modelo. También se incluyen apartes sobre trabajos que detallan la producción de Biodiesel en su parte química, estos permiten conocer las condiciones en las cuales se da dicha producción.

1.4.1 Modelos de fluidos

Los fluidos han sido ampliamente estudiados en mecánica de fluidos y dinámica molecular. Algunos métodos de esos estudios son las ecuaciones de Navier-Stokes (modelos globales) y los métodos de lattice-Boltzmann y lattice-gas [8]. Los métodos de lattice-Boltzmann y lattice-gas tienen un carácter mesoscópico, y se basan en simplificar ecuaciones cinéticas. Estos describen el tiempo de evolución de una partícula que tiene un conjunto discreto de velocidades mientras se mueve en un mallado regular. Nuestra idea se basa en usar método o heurísticas simples de carácter local que se aplica en un mallado regular y que describen la esencia de los procesos envueltos en el mezclado ultrasónico.

1.4.2 Biodiesel

En la Tabla 1 se muestran las propiedades más importantes del Biodiesel desde el punto de vista ambiental, social y físico:

Tabla 1. Propiedades del Biodiesel. Propiedades desde el punto de vista ambiental, social y físico del Biodiesel. III Foro Nacional de Biocombustibles 29 y 30 de Octubre de 2003 (Argentina) (SAGPyA - UCES - Solution Partners)

| |
|--|
| <ul style="list-style-type: none"> • Su producción es renovable. • Su proceso de producción primaria y elaboración industrial determina un balance de carbono menos contaminante que los combustibles fósiles. • Cumple con los requisitos de la <i>Environmental Protection Agency</i> (EPA) para los combustibles alternativos. • Puede emplearse puro o combinado con los combustibles fósiles en cualquier proporción. • No contiene azufre y por ende no genera emanaciones de este elemento, las cuales son responsables de las lluvias ácidas. • Mejor combustión, que reduce el humo visible en el arranque en un 30%. |
| <ul style="list-style-type: none"> • Cualquiera de sus mezclas reduce en proporción equivalente a su contenido, las emanaciones de CO₂, CO, partículas e hidrocarburos aromáticos. Dichas reducciones están en el orden del 15% para los hidrocarburos, del 18% para las partículas en suspensión, del 10% para el óxido de carbono y del 45% para el dióxido de carbono. Estos indicadores se mejoran notablemente si se adiciona un catalizador. • Los derrames de este combustible en las aguas de ríos y mares resultan menos contaminantes y letales para la flora y fauna marina que los combustibles fósiles. • Volcados al medio ambiente se degradan más rápidamente que los petrocombustibles. • Su combustión genera menos elementos nocivos que los combustibles tradicionales reduciendo las posibilidades de producir cáncer. • Es menos irritante para la epidermis humana. • Actúa como lubricante de los motores prolongando su vida útil. |

- Su transporte y almacenamiento resulta 87,0% más seguro que el de los petroderivados ya que posee un punto de ignición más elevado. El Biodiesel puro posee un punto de ignición de 148°C contra los escasos 51°C del gasoil

En [4] se puede encontrar más a cerca de la importancia del Biodiesel como combustible, De [5] se tomaron los datos para hacer los experimentos numéricos bajo las condiciones de producción de Biodiesel, por ejemplo, velocidad del sonido en el medio (1460 m/s) así como también los valores de las presiones presentes en el modelo.

2. ANÁLISIS DEL PROBLEMA

El Biodiesel, es una alternativa de combustible diesel, y es producido con sustancias renovables como lo son el aceite vegetal y grasas animales. Este es biodegradable y produce contaminación apreciablemente menor que la del diesel con respecto al monóxido de carbón, bióxido del azufre, hidrocarburos, partículas y emisiones tóxicas.

En 1997 la producción de combustible Biodiesel fue 550 mil toneladas en Europa, 10,000 toneladas en Malasia y 9,000 toneladas en América del norte. En 2000 la producción anual de combustible Biodiesel en Europa fue 1,210,000 toneladas. Estas cantidades nos dan una idea de la necesidad y la demanda que tiene el Biodiesel.

2.1 Método de producción

El método mas común de producción de Biodiesel es el “proceso de transesterificación alcalina”. En este, dos capas inmiscibles, triglicéridos y metanol reaccionan en la presencia de hidróxido de potasio durante una o dos horas en un reactor produciendo glicerol y esteres metilicos, los cuales pueden ser separados usando gravedad. La capa superior rica en el metilo ácido adiposo Ester, FAM's, se refina utilizando dos extracciones líquidas sucesivas con agua. El producto final es conocido como Biodiesel.

Esta reacción solo ocurre en la región de interfase entre los líquidos y es un proceso bastante lento. Por tanto es requerido tener un mezclado más homogéneo para poder maximizar la superficie de contacto. Masón reportó la hidrólisis de líquidos inmiscibles, aceite y alquenos, utilizando ultrasonido (Albañil, 1999)

2.2 Uso de ultrasonido

El ultrasonido es un proceso de propagación de la compresión (rarefacción) de ondas con frecuencias superiores al rango audible por el humano, i.e. superior 15-16 kHz. (Shutilov, 1988).). Los instrumentos ultrasónicos, comerciales y típicos conocidos como sistemas de probeta tienen un transductor piezoeléctrico accionado por un generador con el que se acopla energía dentro de una reacción química por medio de un transformador de velocidad. (SONICSYSTEMS, 1986)

2.2.1 Por que mezclado ultrasónico

Es claro que la producción de Biodiesel usando mezclado tradicional (mezclado mecánico) es bastante costosa y a la vez necesita mucho tiempo para lograr los productos finales, por tanto el uso de ultrasonido en la producción de Biodiesel es mas que justificado ya que el ultrasonido logra una emulsión bastante fina en la cual la superficie de contacto entre las dos fases es mucho mayor que con el mezclado mecánico y por tanto el tiempo de reacción o “transesterificación” es mucho menor.

En el camino de optimizar el proceso de producción de Biodiesel es necesario obtener condiciones y parámetros que sean bastantes confiables para predecir una producción bastante razonable en el sentido de que no se quiere invertir mas energía de la estrictamente necesaria para obtener el Biodiesel, por que si es cierto que es un combustible renovable también debe ser un combustible económico o cuando menos que su costo sea equiparable con el costo del diesel.

2.2.2 Condiciones y parámetros necesarios

En la producción de Biodiesel es necesario tener en cuenta muchos factores que juegan un papel importante en su producción. Estos factores son: la forma del reactor, las condiciones de temperatura, la naturaleza del estímulo ultrasónico, las posiciones y cantidades de transductores, la cantidad o razones de las fases, la proporción de cavidades iniciales en los líquidos y muchos otros factores que es necesario estudiar para poder encontrar las condiciones optimas de tal forma que la producción tenga un costo que pueda competir y sea asequible al usuario. Algunos de los parámetros anteriormente mencionados se variaron durante las pruebas del modelo con el fin de obtener los mejores valores que permiten optimizar el proceso de mezclado.

2.3 Modelo general

El modelo se desarrolló sobre un mallado regular aplicado a dos fluidos o capas inmiscibles, en el que se supone la existencia de cavidades, dicha existencia está sujeta a

distribuciones que dependen de las condiciones del modelo. En un lugar en ese mallado se asume la existencia de un punto, que se usa para simular la presencia de un transducer. Este punto nos sirve para referirnos al lugar de emisión del ultrasonido.

El comportamiento de las cavidades es descrito por un modelo [3] que se basa en la ecuación de Rayleigh–Plesset [6]. Este modela el comportamiento del radio R y la energía E (en función del tiempo) de una burbuja a partir de un radio inicial r_o , de una frecuencia f y una amplitud de presión a . Claro está que los parámetros r_o , f y a , por las condiciones en las que se desarrolla el modelo, están sujetos a variaciones que dependen en gran parte de las condiciones alrededor de cada cavidad y de la posición de referencia para la emisión del ultrasonido.

A continuación describimos el comportamiento de una cavidad hasta el momento de la implosión, momento en el cual aproximamos la energía despedida por la cavidad y con esta estimación aproximamos la masa que movemos alrededor de cada cavidad.

Para desarrollar este modelo nos basamos en cuatro suposiciones que tienen que ver directamente con la forma del sistema.

- La primera suposición se refiere la descripción de la vida de la cavidad, esta es dada por un modelo basado en la ecuación de Rayleigh–Plesset [6].

- La segunda se refiere a la presencia de las cavidades “semillas”; estas se asumen ya que el modelo que tenemos tiene como condición un radio inicial.
- La tercera un factor de variación en el estímulo que pueda predecir o aproximar la interacción del ultrasonido con el medio.
- La cuarta es sobre el intercambio de masa que ocurre después de la implosión y que dependerá de una o más direcciones de flujo.

La implementación del modelo es muy versátil, y estas suposiciones pueden variarse fácilmente, para usar el modelo como un laboratorio virtual. Nuestro deseo ante todo es reflejar en las simulaciones la esencia del mezclado. El modelo servirá entonces como un laboratorio virtual, donde en primera instancia se deben comparar las simulaciones con los experimentos, y en segunda instancia (una vez ajustada la simulación a la realidad) se pueden probar distintas opciones de diseño a muy bajo costo.

Nosotros hemos asignado algunas restricciones o condiciones dentro del sistema. Estas condiciones pueden ser variadas para ver más simulaciones y para ajustar a la realidad.

- El reactor es infinito (no tenemos paredes, las ondas viajan indefinidamente), ya que la cantidad del volumen que estamos modelando nos permite asumir esto.

- La interacción entre dos cavidades o en general entre cavidades no es tomada en cuenta.
- La reflexión de las ondas en las paredes del reactor son despreciadas por tratarse de un reactor infinito.

Como se vio anteriormente dentro del modelo se incluye parámetros de radio y energía que describen la vida de una cavidad, estos parámetros son usados para dar una representación a escala del tamaño (radio) de la cavidad a lo largo de toda su vida (tiempo antes de implotar) y dar una tonalidad a las cavidades que dependa de la energía que contienen las cavidades en cada instante de tiempo. Las capas de los líquidos son representadas como dos fases de diferentes colores y se usa transparencia en los colores para ver la interacción entre las capas.

2.3.1 Cavitación, Implosión y Ultrasonidos

La **cavitación** es la formación y la acción de cavidades que en nuestro caso están llenas de gas en el interior de un líquido expuesto a un intenso campo ultrasónico. En un medio líquido, las ondas ultrasónicas generadas por un transductor, crean ondas de compresión y depresión a altísima velocidad. Estas ondas de compresión y depresión en el líquido originan el fenómeno de la "cavitación". Durante la primera fase, la depresión, se crean en el interior del líquido una infinidad de burbujas (cavidades). Durante la segunda fase, la compresión, la enorme presión ejercida sobre estos puntos hacen que

estos se compriman hasta hacerlos colapsar en sí mismos (*implosión*) o simplemente acumular energía para volver a pasar a la primera fase; con la consiguiente despedida de energía que a su vez imprime una velocidad o flujo dentro del fluido produciendo un movimiento de masa. Por otro lado, dado que se está estudiando el comportamiento de la burbuja a partir de su radio, definimos entonces el *fenómeno de implosión* como el instante en el cual la velocidad de la pared de la burbuja alcanza la velocidad del sonido en el líquido y forma así una discontinuidad o “*shock*”. La velocidad del sonido que se ha considerado es de 1460 m/seg según las condiciones dadas en la producción de Biodiesel [5]. En cuanto al ultrasonido, se trata de ondas sonoras de frecuencia superior al límite máximo de audibilidad (16 kHz); las frecuencias máximas con las cuales se harán los experimentos numéricos serán de hasta 2 o 3 MHz.

2.3.2 Ecuación de Rayleigh-Plesset

La ecuación isotérmica y viscosa esta dada por:

$$\rho \left[R\ddot{R} + \frac{3}{2}\dot{R}^2 \right] + 4\mu \frac{\dot{R}}{R} = \left(p_0^\infty - p_v + \frac{2\sigma}{ri} \right) \left(\frac{ri}{R} \right)^3 + p_v - \frac{2\sigma}{R} - p_0^\infty + a \sin(ft) \quad (2.1)$$

Esta modela el comportamiento del radio R de una burbuja a partir de un radio inicial ri , de una frecuencia f y una amplitud de presión a . El resto de las constantes que aparecen en la ecuación se describen a continuación:

ρ : Densidad del líquido.

μ : Viscosidad.

σ : Tensión superficial del líquido.

p_v : Presión de vapor.

p_0^∞ : Presión ambiental.

Esta ecuación se denomina isotérmica por que se presume una temperatura constante y se le denomina viscosa, por que a diferencia de otras versiones de la ecuación Rayleigh-Plesset aparece el término $4\mu \frac{\dot{R}}{R}$, donde μ representa viscosidad.

En el Apéndice se puede ver como se usa esta ecuación para construir la función que describa la vida de una cavidad.

2.3.3 Cavidades (semillas).

En el modelo dado por 2.3 se puede apreciar claramente que la ecuación usa un radio inicial r_i , por lo tanto para nuestro modelo estamos sujetos a suponer la existencia de cavidades iniciales que tienes un radio inicial dado, para esto asumimos que en el fluido existen cavidades (semillas) sobre la cuales actúa el ultrasonido, esta existencia es basada en la idea de que la mayoría de los fluidos contienen gases, impurezas o partículas y estos pueden iniciar la aparición de cavidades, para nosotros esta aparición puede ser simulada por una distribución de probabilidades (sección 3.2).

2.3.4 Variación del ultrasonido.

El ultrasonido, como toda onda, cuando cambia de medio se difracta o refleja y esto lo podemos medir en función de una variación, por lo tanto es necesario contar con la

posibilidad de variar el ultrasonido dependiendo del medio o el sistema donde este se este aplicando. Esta variación puede ser aproximada usando un factor que dependa exclusivamente de los elementos que el ultrasonido atraviesa.

2.3.5 Intercambio de masa.

En el momento en que una cavidad completa su ciclo de vida, esta implota y libera energía, gran parte de esta energía se convierte en un “jet” o chorro que actúa sobre cada una de las partículas de los líquidos inmiscibles que se encuentran alrededor o en una vecindad de la cavidad. Esto produce un flujo local que realiza un movimiento de partículas dentro del volumen y consecuentemente este flujo crea un mezclado bastante homogéneo y mejor que el producido por el mezclado tradicional. Para modelar este intercambio nosotros usamos una idea de intercambio sencilla que conserva la masa y obedece a una dirección de flujo aleatoria (sección 3.4.4).

3. EL MODELO

El modelo que desarrollamos a continuación se basa en resolver el problema de manera microscópica (local) tomando a los fluidos como una gran cantidad de diminutas partículas. Para esto utilizaremos un mallado regular en el espacio \mathbb{R}^3 (3D regular grid), donde el tiempo y el estado de cada celda son también discretos y cada partícula del mallado tiene 18 direcciones en las que se puede mover. La figura 1 nos da una idea de como se puede mover una partícula en este sistema.

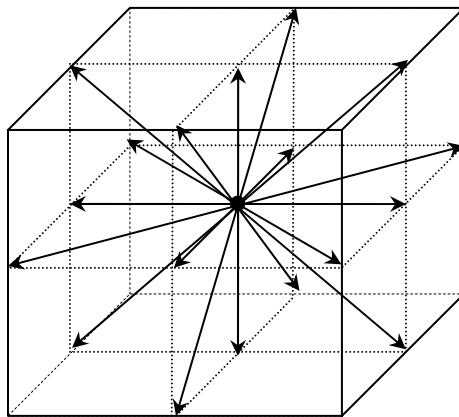


Figura 1. Geometría del mallado. Las 18 posibles direcciones son mostradas como flechas.
 $\{[x \ y \ z]/x,y,z \in \{-1,1,0\} \text{ y al menos uno es } 0\} - \{0,0,0\}$

3.1 Modelo estático

Este modelo supone la existencia de una función que modela el radio, energía y vida (tiempo antes de implotar) de una burbuja o cavidad a partir de un radio inicial, frecuencia, amplitud y estado del sistema al momento de la *inceptión* de la cavidad en el sistema. Este modelo estático es basado en la idea de modelar el comportamiento de una

cavidad suponiendo que los parámetros iniciales son *estáticos*, es decir, no son variados durante todo el periodo de tiempo que conforma la vida de la cavidad. La función que suponemos debe ser capaz de determinar toda la información de radio, energía y tiempo de vida a partir de esos parámetros.

Cada punto en el mallado es usado como referencia para la posible aparición de una cavidad. Esta *incepción* está sujeta a una distribución aleatoria de semillas, esta distribución decide la existencia de una cavidad en cada posición del sistema en el momento que esa posición este libre (no tengamos una cavidad ese lugar). En Figura 2 se puede ver una idea del modelo que se propone para la distribución de las cavidades dentro del sistema.

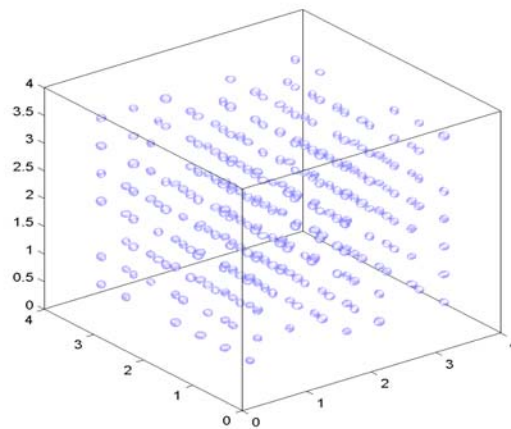


Figura 2. Geometría del sistema. Una mallado que contiene semillas distribuidas con una probabilidad $p=0.5$ de *incepción*.

3.2 Suposiciones del modelo

1. (S1) Existe una función $(tv, \vec{R}, \vec{E}) = F(ri, f, a, \vec{R}_x)$ donde ri es el radio inicial de la burbuja, f y a son la frecuencia y la amplitud inducida en la burbuja, \vec{R}_x es un factor que depende del estado del sistema al momento de aplicar la función, \vec{R} y \vec{E} son los vectores resultantes que describen discretamente el radio y la energía de la burbuja en el tiempo de vida tv .
2. (S2) Existe una base de cavidades iniciales (semillas) determinadas por una “*distribucion*” de probabilidades.
3. (S3) Supondremos que la intensidad con la que el ultrasonido ataca a una cavidad c_x está sujeta a una variación que depende de la “*distribución*” de cavidades que existen entre esta cavidad c_x y el transducer.
4. (S4) Supondremos que la mezcla se realizara dependiendo de la cantidad de energía despedida por las cavidades y con un intercambio simple.

Hasta este momento hemos definido las reglas del juego para nuestro modelo. Solo resta determinar concretamente como se trabaja con cada una de estas hipótesis.

3.3 Construcción del modelo

En esta parte nos dedicaremos a describir la forma en la que se comportaran las cavidades y la existencia o no, de cada una de ellas en un nodo. Particionaremos el espacio con un mallado regular (3D regular *grid*) con una partición de n^3 cajas en donde el tiempo y el estado de cada partícula son discretos.

3.3.1 Distribución (S2)

Como parte inicial tenemos una distribución de cavidades dada por $S2$ en todo tiempo t . Esta distribución se asignara de forma tal que la probabilidad de que una cavidad este presente es $p = 0.3$ (esta probabilidad puede ser variada). Podemos decir que esta distribución asigna valores a una variable aleatoria $p(\mathbf{x}, t) \in \{0, 1\}$ que indica la presencia (1) o ausencia (0) de una cavidad en el nodo \mathbf{x} .

En el tiempo cero ($t=0$) asignamos valores a $p(\mathbf{x}, t)$ para cada uno de los elementos del mallado, este para nosotros es nuestro grupo llamado “semillas”, para los demás tiempos usamos el siguiente criterio.

El valor de $p(\mathbf{x}, t)$ siempre es 1 mientras $(t_i + tv_x^i) - t > 0$ (sección 3.3.4), esto es, mientras la vida de la cavidad no haya terminado. En el momento en que la vida de la cavidad termina $p(\mathbf{x}, t)=0$, nuevamente se tiene que usar la variable aleatoria para

determinar el valor de $p(\mathbf{x}, t+1)=0$ para decidir la presencia o ausencia de la cavidad, esto es

$$p(\mathbf{x}, t) = \begin{cases} R(p) & \text{si } p(\mathbf{x}, t-1) = 0 \\ 1 & \text{si } p(\mathbf{x}, t-1) = 1 \text{ y } (t_i + tv_x^i) - t > 0 \\ 0 & \text{otro caso} \end{cases} \quad (3.1)$$

Donde t_i es el tiempo donde se cálculo \mathbf{F} (ecuación 3.5) por última vez, tv_x^i es uno de los parámetros que se obtuvieron como resultados en \mathbf{F} y $R(p)$ es una variable aleatoria de Bernoulli, que vale 1 con probabilidad p y 0 con probabilidad $1-p$.

Para el uso de \mathbf{F} tenemos que realizar los siguientes cálculos. La distancia d de la cavidad c_x al transductor, para esto usaremos la medida de distancia euclidiana en \mathbb{R}^3 y una constante \bar{R}_x , para el cálculo de esta utilizaremos una medida para la “distribución” de cavidades ($S3$) en un estado de tiempo en el entorno de una cavidad.

3.3.2 Conjunto C_x .

Para cada punto \mathbf{x} se define una vecindad enmarcada por un cono con un ángulo θ , esto es, dado un punto en el espacio \mathbf{x} se construye un cono con su interior (conjunto C_x) que comience en el lugar donde se encuentra el transductor y termine en el punto \mathbf{x} (ver Figura 3).

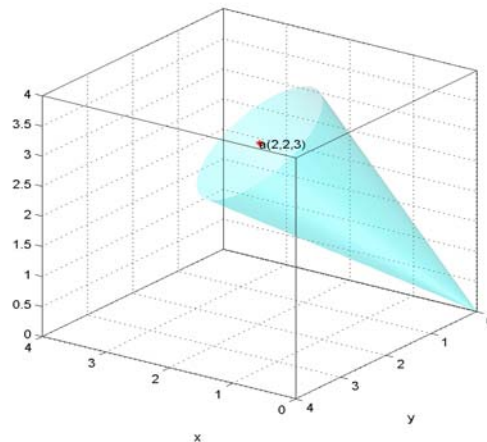


Figura 3. Conjunto C_x , el transducer se encuentra en el punto $(0, 0, 0)$ y la cavidad c_x se encuentra en el punto $x(2,2,3)$ del mallado.

Una forma de construir el conjunto C_x es la siguiente:

Supongamos que el transducer se encuentra el punto $\mathbf{p}(\mathbf{a}, \mathbf{b}, \mathbf{c})$, para un punto \mathbf{y} , este pertenece a C_x si cumple:

- I) El ángulo entre el vector $\mathbf{p-x}$ y el vector $\mathbf{p-y}$ es menor o igual que $\theta/2$ y mayor que o igual que 0 grados esto es

$$\cos(\theta/2) \leq \frac{(\mathbf{p-x}) \cdot (\mathbf{p-y})}{\|\mathbf{p-x}\| \|\mathbf{p-y}\|} \leq 1 \quad (3.2)$$

- II) La proyección del vector $\mathbf{p-y}$ sobre el vector $\mathbf{p-x}$ debe ser menor que la norma del vector $\mathbf{p-x}$ esto es

$$(\mathbf{p-x}) \cdot (\mathbf{p-y}) \leq \|\mathbf{p-x}\|^2 \quad (3.3)$$

- III) La distancia entre el punto \mathbf{x} y el punto \mathbf{y} debe ser menor que la distancia entre el punto \mathbf{p} y el punto \mathbf{x} esto es

$$\|(\mathbf{x-y})\| \leq \|\mathbf{p-x}\| \quad (3.4)$$

3.3.3 Factor \bar{R}_x

El factor \bar{R}_x en el tiempo t para la cavidad c_x está determinada por

$$\bar{R}_x = \frac{\sum_{y \in C_x} p(y, t-1)}{n(C_x)} \quad (3.4)$$

donde $n(C_x)$ es el número de elementos de C_x .

Claramente esta constante está entre 0 y 1, Esta constante describe la dispersión de cavidades antes de un punto x , entre mas cerca este de 1, mas lleno de burbujas es el espacio dentro del entorno C_x y por tanto hay una mayor variación en el ultrasonido que llega al punto x .

Habiendo definido claramente como se distribuyen las cavidades dentro del fluido y cual es la constante de variación en su entorno, pasemos entonces a desarrollar el método que utilizaremos para implementar la función dada en (SI).

3.3.4 Uso de la Función F (SI)

La función F tomada para (SI) es dada por un modelo con base en la ecuación isotérmica y viscosa de Rayleigh-Plesset [6]. Esta modela el comportamiento de la burbuja a partir de un radio inicial ri , de una frecuencia f y una presión de una amplitud a , En el apéndice se explica como se usa esta ecuación para construir F.

Para aplicar la función $(tv, \bar{\mathbf{R}}, \bar{\mathbf{E}}) = \mathbf{F}(ri, f, a, \bar{\mathbf{R}})$ en un nodo x donde exista una semilla, tenemos que verificar que $p(x, t_i)=1$. Los parámetros iniciales r_i, f y a son asignados de antemano, luego calculamos el factor $\bar{\mathbf{R}}_x$ para la cavidad c_x y aplicamos la función \mathbf{F} con los siguientes parámetros

$$(tv_x^i, \bar{\mathbf{R}}_x, \bar{\mathbf{E}}_x) = \mathbf{F}(ri, f, a, \bar{\mathbf{R}}_x) \quad (3.5)$$

en donde el parámetro $\bar{\mathbf{R}}_x$ se usa para cambiar la frecuencia, la amplitud o ambas. En este trabajo es usado para variar la amplitud a , esto es, a es remplazada por $a \bar{\mathbf{R}}_x$. Los valores resultantes son:

- Tiempo de vida tv_x^i ; este valor está dado en unidades de Δt y nos dice cuantas unidades de tiempo Δt tenemos que asignar para la vida de esta cavidad.
- Radio $\bar{\mathbf{R}}_x$; este vector contiene tv_x^i+1 elementos que describen el comportamiento del radio de la cavidad c_x en cada instante durante un tiempo tv_x^i de vida y asigna los valores del radio de la cavidad en función del tiempo de la siguiente forma

$$r(x, t_i : t_i + tv_x^i) = \bar{\mathbf{R}}_x$$

- Energía $\bar{\mathbf{E}}_x$; este vector contiene $tv_x^i + 1$ elemento que describen el comportamiento de la energía ganada por la cavidad c_x a lo largo del tiempo tv_x^i de vida y asigna los valores de energía de la cavidad en función del tiempo de la siguiente forma

$$E(x, t_i : t_i + tv_x^i) = \bar{\mathbf{E}}_x$$

Esta asignación nos dice todo sobre la cavidad que existe en el nodo x durante un periodo de tiempo tv_x^i a partir de un tiempo t_i . Cuando la cavidad agota su tiempo de vida tenemos que $p(x, t_i + tv_x^i) = 0$ y $p(x, t_i + tv_x^i + 1) = R(p = \lambda)$ y si el resultado es 1, se realiza la *inceptión* de una nueva cavidad y nuevamente se repite el cálculo del valor \bar{R}_x y se aplica **F** en la misma forma. Es decir, este proceso solo se ejecuta en el primer tiempo $t_b \geq t_a$ en el que $(t_i + tv_x^i) - (t_a - 1) = 0$ y $p(x, t_b) = 1$ y en este momento asignamos $t_{i+1} = t_b$.

3.4 Volumen

Para representar el volumen estamos suponiendo que tenemos un reactor infinito y solo tomamos una parte del reactor, en la que el transductor se encuentra en el centro. Es claro que la posición del transductor puede ser modificada.

3.4.1 Mallado del volumen

El volumen es dividido en n^3 cajas, en donde cada caja representa una partícula de fluido. Para representar numéricamente las cajas asignamos 1 ó 0 para cada partícula indicando que tipo de fluido está representando la partícula.

El tamaño de la discretización del volumen dependerá del radio inicial que se le de a las cavidades y de que tan buena aproximación queremos del movimiento que se está

realizando, además también depende de la cantidad de cavidades o del mallado que se usa para generar las cavidades.

Sea Δs la variación de espacio que usaremos en las tres direcciones x , y , z , entonces tenemos que el volumen de un partícula es Δs^3 . Por lo tanto la masa de la partícula $m_p = d * \Delta s^3$ donde d es la densidad.

3.4.2 Mezclado.

Con el modelo dado anteriormente podemos estimar el momento en que implota y la cantidad de energía que expulsa cada cavidad; esto nos dará el momento y la cantidad de masa que hay que mover alrededor de cada cavidad.

Para calcular la cantidad de masa que hay que mover alrededor de una cavidad tenemos que hacerlo a partir de la energía expulsada por la cavidad y la condición de que la cavidad implota cuando la pared de esta se mueve a la velocidad del sonido.

3.4.3 Cálculo de la masa

El cálculo de la masa está dado por la energía E en el momento que implota la cavidad, la velocidad del sonido v_s y una constante α de pérdida de energía. Esta pérdida puede depender de los parámetros iniciales y además por la energía convertida en calor

$$\begin{aligned} \alpha E &= \frac{1}{2} m v_s^2 \\ m &= \frac{2\alpha E}{v_s^2} \end{aligned} \quad (3.6)$$

La masa dada por las formula anterior es la masa total a mover alrededor de la cavidad y es la cantidad que nos permite saber cuantas partículas de los fluidos son las que movemos.

3.4.4 Partículas

Las partículas que se deben de mover dependen completamente del mallado que se está utilizando. Como cada partícula va a tener una masa que depende exactamente del mallado que se este utilizando asumamos que tenemos partículas con masa m_p y entonces la cantidad de partículas que se deben de mover está dada por la razón

$$n = \frac{m}{m_p}. \quad (3.7)$$

n es el número de partículas a mover y decidimos realizar ese movimiento dependiendo de un intercambio sobre una dirección (ver Figura 4)

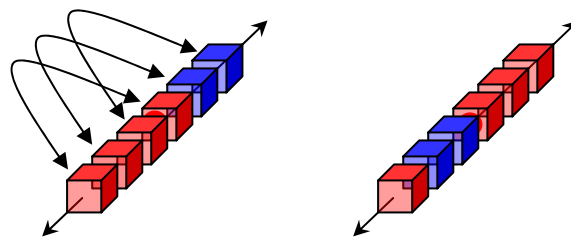


Figura 4. Intercambio de seis partículas en una dirección de las mencionadas en la figura 1.

Debemos de estar de acuerdo que este es el esquema más simple de intercambio de masa que podemos usar. Podemos usar cualquier otro esquema mas complicado, tan

complicado como se desee, por ejemplo, podríamos usar un modelo de colisión y propagación.

3.5 Descripción del algoritmo

3.5.1 Esquema general

El modelo que se plantea anteriormente es un modelo en el que la mayoría de los procesos son locales y no tenemos que resolver ningún tipo de sistema global (acoplado), por lo tanto los procesos se pueden realizar uno a uno en un orden específico sin que esto afecte los cálculos, como se ve en el modelo cada cálculo solo depende del tiempo anterior y cada nuevo resultado se guarda independiente de los demás resultados.

Para la descripción precisa del algoritmo que usa el modelo tenemos que estar de acuerdo en cada uno de las suposiciones y condiciones anteriores.

1. El modelo usa como primera medida la suposición (**S2**), con el asigna valor de 0 o 1 a las posiciones de una matriz \mathbf{A}_t (o simplemente \mathbf{A}) que representa las semillas en el tiempo 0.
2. Usamos la suposición (**S3**) para calcular el factor \bar{R}_x y aplicamos la función \mathbf{F} para calcular la descripción del radio R , la energía E y el tiempo de vida tv de cada cavidad. Esto se hace en las posición donde \mathbf{A}_t tiene el valor de 1, tenemos

que recordar que \mathbf{F} depende grandemente de la condiciones del medio en el que se está trabajando.

3. En cada posición (i,j,k) en la que $\mathbf{A}_t(i,j,k)$ es 0 usamos la suposición (S2) para asignar valor de 0 o 1 a la posición en la matriz en el siguiente instante de tiempo $\mathbf{A}_{t+1}(i,j,k)$. Si el valor es 0 no existe cavidad y solo pasamos a la siguiente posición de la matriz, si el valor es 1 entonces tenemos que usar la suposición (S3) para calcular el factor \bar{R}_x y aplicamos la fusión \mathbf{F} para calcular la descripción del radio R , la energía E y el tiempo de vida t_v de esta cavidad.
4. En el instante en que una cavidad ha terminado su ciclo de vida el valor para $\mathbf{A}_t(i,j,k)$ es 0 y por tanto tenemos que aplicar nuevamente 3.

Esta parte solo nos ha descrito como dar vida a las cavidades; nos resta entonces describir como se usan estos cálculos para realizar un mezclado.

Suponiendo que tenemos el volumen dado en 3.5.

5. En cada instante de tiempo el que una cavidad implota debemos tomar la energía desprendida por esa cavidad y aplicando 3.4.2 y 3.4.3 realizamos los movimientos locales de las partículas.

3.5.2 El algoritmo secuencial

El algoritmo se conforma básicamente de cuatro matrices tridimensionales A , T_i , T_v y MA . La primera es para identificar la existencia de una cavidad en una posición ijk del mallado, la segunda se encarga de registrar los tiempos en los que inicia la vida de una cavidad, la tercera se encarga de registrar los tiempos de vida (instante de vida) de cada cavidad y la cuarta es la matriz que representa el volumen con los líquidos.

Cuando una cavidad nace o es insertada se calcula su vida, cuando esta muere se hace un mezclado. Para representar esto nos basamos en tres matrices RL , EL y MAT 4 dimensiones (3 dimensiones para representar el volumen y una para representan el tiempo), cada una de estas matrices registran el radio, la energía y el mezclado a medida que pasa el tiempo.

En la figura 5 se puede ver una descripción del algoritmo, las funciones mas importantes de este algoritmo son dos, la primera es la función **variación**, la cual es construida con base a la sección 3.3.3 y la segunda es la función **rayle** (sección 3.3.4) que es el resultado de solucionar Rayleigh-Plesset [6], la cual utiliza otras dos funciones **rigid** y **estimulo**. El código C para el programa se encuentra en el apéndice 2.

```

1 Semillas(A,P);
2 Volumen_inicial(MA,R);
3 for(i=0;i<N;i++)
4     for(j=0;j<N;j++)
5         for(k=0;k<N;k++)
6             if (A[i][j][k]==1)
7                 {
8                     Rp=variación(A,i,j,k,theta);
9                     tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,t);
10                }
11
12 for(ta=1;ta<=TIEMPO;ta++)
13 {
14     for(i=0;i<N;i++)
15         for(j=0;j<N;j++)
16             for(k=0;k<N;k++)
17                 {
18                     if (A[i][j][k]==0)
19                         {
20                             Rp=variación(A,i,j,k,theta);
21                             U=random(P);
22                             if (U<P)
23                                 {
24                                     tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,f,a,ta);
25                                     Ti[i][j][k]=ta;
26                                     Tv[i][j][k]=tv;
27                                     B[i][j][k]=0;
28                                 }
29                         }
30                     else
31                         if (Ti[i][j][k]+ Tv[i][j][k]==0)
32                             {
33                                 dir=randir();
34                                 E=EL[i][j][k][ta];
35                                 Numpar=particulas(mp,E,Vs);
36                                 mezcla(MA,i,j,k,Numpar,dir);
37                                 B[i][j][k]=0;
38                             }
39                 }
40     actualiza(A,B);
41     guardarM(MAT,MA,ta);
42     VM[ta]=medidaM(MA);
43 }

```

Figure 5. El algoritmo secuencial.

Descripción de las funciones

Volumen_inicial(MA,R)

Esta función recorre toda la matriz MA llenado con ceros y unos dependiendo de los porcentajes iniciales que se quiera dar de los fluidos.

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
      if (k < N*R)
        MA[i][j][k]=1; /*liquido uno*/
      else
        MA[i][j][k]=0; /*liquido dos*/
```

Esto está representando un volumen con dos capas,

La complejidad de esta función es $O(N^3)$.

Semillas(A,P)

Esta función recorre toda la matriz A llenado con ceros y unos dependiendo de la condición de probabilidad dada.

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
      U=rand;
      if (U<P)
        A[i][j][k]=1; /*existe una cavidad*/
      else
        A[i][j][k]=0;
```

Esto está representando un mayado con un porcentaje de cavidades inmersas.

La complejidad de esta función es $O(N^3)$.

Rp=variación(A,i,j,k,theta)

Esta función recorre toda la matriz A encontrado todos los elementos que pertenezcan a un conjunto $C(i,j,k,theta)$ (figura3) y calcula la razón

$$R_p = \frac{\text{suma de } A(i, j, k) \in C(i, j, k, \theta)}{\text{numero de elementos de } C(i, j, k, \theta)}$$

La complejidad de esta función es $O(N^3)$.

```
tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,f,a,t);
```

Esta función calcula la vida de una cavidad, esto se hace solucionando numéricamente la ecuación diferencial Rayle- Plesset usando el conocido método Runje Kutta de orden 4 una descripción de esta función se encuentra en el apéndice 2.

La complejidad de esta función es $O(1)$.

randir()

Es una función que escoge aleatoriamente un elemento del conjunto

$$\{[x \text{ y } z]/x,y,z \in \{-1,1,0\} \text{ y al menos uno es } 0\} - \{0,0,0\}$$

La complejidad de esta función es $O(1)$.

partículas(mp, E, Vs)

Esta función calcula la cantidad de partículas que se moverán alrededor de la cavidad representada por $A(i,j,k)$, esto se hace con el siguiente cálculo

$$\begin{aligned} mp &= \text{densidad} * \text{volumenP}; \\ \text{masa} &= 2 * E / Vs^2; \\ \text{Numpar} &= \text{masa} / mp; \end{aligned}$$

La complejidad de esta función es $O(1)$.

mezcla(MA, i, j, k, Numpar, dir)

Esta función hace el intercambio de **Numpar** partículas alrededor del punto **ijk** de **MA** en la dirección **dir** (ver figura)

```
for(n=0;n< Numpar;i++)
{
  r1=dir*n;
  r2=dir*(n-Numpar);
  aux=MA[x+r1[1]][y+r1[2]][z+r1[3]];
  MA[i+r1[1]][j+r1[2]][k+r1[3]]=M[i+r2[1]][j+r2[2]][k+r2[3]];
  MA[i+r2[1]][j+r2[2]][k+r2[3]]=aux;
}
```

La complejidad de esta función es $O(N)$.

actualiza(A,B);

Esta función recorre la matriz **A** reemplazando sus valores por los de **B**

La complejidad de esta función es $O(N^3)$.

guardarM(MAT,MA,ta)

Esta función recorre la matriz **MAT** (en la posición **ta**) reemplazando sus valores por los de **MA**;

La complejidad de esta función es $O(N^3)$.

medidaM(MA)

Esta función recorre la matriz **MA** y en cada elemento realiza un cálculo sobre los elementos que están alrededor.

```
vermT=0;
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
```

```

{
verm=0;
for ii=-1:1
  for jj=-1:1
    for kk=-1:1
      if (coolordif(i,j,k,ii,jj,kk))
        verm=verm+1;
      verm=verm/26;
      vermT=vermT+verm;
    }
  vermT=vermT/pow(N,3);
}

```

La complejidad de esta función es $O(N^3)$.

3.5.3 Complejidad del Algoritmo

Si miramos el algoritmo de la figura 5 podemos decir lo siguiente.

Las funciones **semillas** y **variación** requieren tiempo $O(N^3)$ cada una. El resto del algoritmo consiste de dos bucles principales.

El primero consiste de tres bucles anidados, cada uno de largo N . El cuerpo del bucle interior se domina por **variación** la cual tiene orden $O(N^3)$, así que el primer bucle principal requiere tiempo $O(N^6)$.

El segundo bucle es de largo T y su cuerpo está dominado por un triple bucle que es también de orden $O(N^6)$. Por tanto el algoritmo, requiere tiempo $O(TN^6)$.

Hasta el momento T ha sido usado como un parámetro arbitrario.

3.5.4 El uso del Paralelismo

La cantidad de cálculo y memoria requerida para sistemas grandes sugiere que utilicemos las ventajas de implementar el problema en forma tal que lo podamos correr en varios procesadores.

La ventaja de los modelos construidos basados en conceptos locales y estáticos permite que la paralelización de los proceso sea lo más satisfactoria posible, es decir, lograr tener una aceleración lineal al momento de incrementar el numero de procesadores.

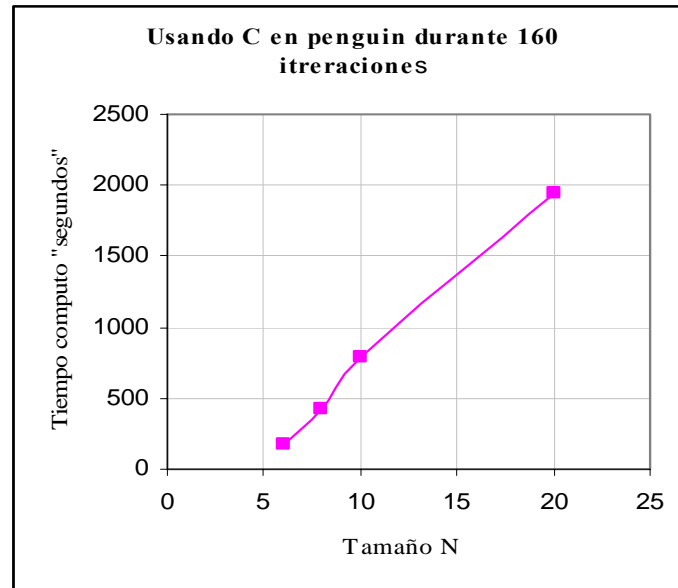
Nuestro modelo se basa en ideas que en primera instancia son locales y solo dependen del paso anterior para ser desarrolladas.

El tiempo de CPU y memoria requeridas por este modelo es de orden no mayor que $N^D \times N^D$ en dada intervalo de tiempo T, donde N es tamaño del sistema, D es la dimensión del espacio, considerando que el sistema se resuelve en el tiempo tenemos un problema de gran escala.

La tabla 2 y figura 6 muestran los tiempos de cómputo de algunos ejemplos, en donde se ve el aumento progresivo en tiempo de computo, al pensar en querer resolver este problema con un tamaño de 128^3 durante 500 o 1000 iteraciones, lo mejor es trabajar en paralelo.

Tabla 2. Tiempos de computo algoritmo en C y algoritmo en Matlab

| N^3 | Número de pasos | Matlab | C |
|--------|-----------------|-----------|------------|
| 6^3 | 160 | 630.56 s | 177.897 s |
| 8^3 | 160 | 1797.04 s | 419.603 s |
| 10^3 | 160 | 2897.30 s | 783.133 s |
| 20^3 | 160 | 7862.56 s | 1951.037 s |

**Figure 6.** Instantes de tiempo vs razón numérica que representa el mezclado.

3.6 Método de Paralelizacion

El plantear este modelo como estrategias de trabajo local nos permite implementar una versión paralela del algoritmo sin muchas alteraciones en el código secuencial. Dentro de las técnicas utilizadas para paralelizar nosotros consideramos “maestro-esclavo” pero por la naturaleza del algoritmo esta técnica con asignación de trabajo

dinámico resulta ineficiente por tratarse de un volumen en el que a los elementos de este se le aplican operaciones similares. Resulta mejor la idea de paralelizar usando asignación de trabajo estático, en esta técnica nos dedicamos a distribuir el volumen en forma “casi” simétrica sobre cada uno de los procesos.

Para la repartición de trabajo se realizó una partición del volumen a lo largo del uno de los ejes.

El criterio usado es el siguiente:

Supongamos que tenemos un volumen de tamaño N y tenemos M procesos entonces cada proceso toma $N \times N \times (\text{int}(i \times N/M) : \text{int}((i+1) \times N/M))$ (suponiendo que las matrices tienen N elementos en cada fila “0:N-1”) parte del volumen, donde el parámetro i representa el **rank** del proceso. Esta estructura así diseñada no es a prueba de fallos. Debido a que cada procesador tiene una cantidad N^3/NP aproximadamente del volumen y por tanto es irremplazable (ver figura 7), en esta figura se muestra con flechas rojas el intercambio de información en la frontera de partición del volumen MA y las demás solo indican la actualización de la matriz A (matriz de cavidades).

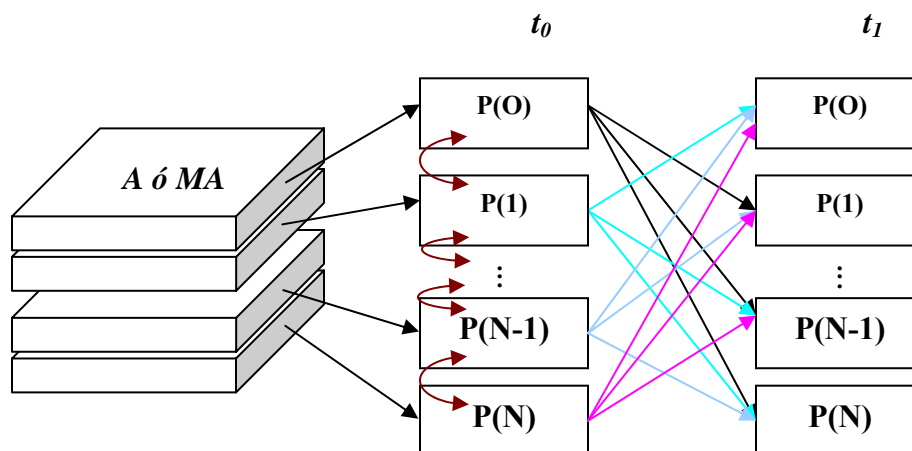


Figura 7. Esquema de la técnica usada

3.6.1 Implementación del Algoritmo

Acerca de sobre MPI (*Message Passing Interface*)

MPI es una librería de funciones, la cual puede ser usada con los lenguajes C, o Fortran. Está basada en “paso de mensajes”, una forma eficiente y cómoda de programar sistemas en paralelo. Una de las ventajas más notables de la utilización de MPI es su implementación y portabilidad a una gran variedad de plataformas, desde máquinas con memoria compartida (como la Origin2000) hasta redes de estaciones de trabajo (como un “Cluster Linux”). Cuando se ejecuta un programa en MPI, cada procesador tiene una copia del programa. Todos los procesadores inician simultáneamente el mismo listado del programa, pero cada proceso ejecutará distintas instrucciones del programa.

3.6.2 El algoritmo con MPI

A continuación se presenta una breve descripción del código utilizado para implementar la simulación utilizando MPI. El código completo se encuentra en el apéndice 2. Prácticamente, el programa en paralelo solo usa unas cuantas variantes del programa secuencial.

El programa secuencial consta de tres partes importantes, la creación de la “semillas”, vida de las semillas y el desarrollo de las demás cavidades, y mezclado en los siguientes instantes de vida.

El algoritmo en paralelo usa esa tres partes, todo esto acompañado de la partición del **for** principal del mallado que corresponde a las cavidades.

Las librerías de funciones que se usan en el algoritmo en paralelo son las mismas usadas en el secuencial, las diferencias más sobresalientes son que los cálculos que corresponden a los elementos que se encuentran en la fronteras de división del volumen, estos son guardados y realizados con intercambio de información de los procesos que contienen las partes del volumen que se envuelven en estas operaciones.

3.6.3 La implementación

Para implementar el modelo planteado en paralelo procederemos de la siguiente forma. Toda nuestra descripción se basará en el algoritmo dado en la figura 5 y la distribución de trabajo se hace usando la partición dada en 3.6.

Los cálculos de las funciones dadas en la línea 1 y 2 se hacen usando paralelo y seguido de esto cada procesador se encarga de enviar la parte de la matriz A que le corresponde a los demás procesos.

El proceso de la línea 3 a 10 se realiza para encontrar la vida de las cavidades, esta parte se realiza partiendo el primer **for** usando la partición 3.6.

En los siguientes pasos esta el cálculo de los demás instantes de tiempo y nuevamente en la línea 14 usamos la partición 3.6 pero el cálculo de línea 36 solo se aplica para los elementos que no están en la frontera de la partición, los que se encuentra en la frontera se guardan y se calculan al final de cada iteración.

Para realizar el cálculo de la línea 36 en los elementos de frontera se realiza dos a dos entre los procesos como se ve en la figura 8.

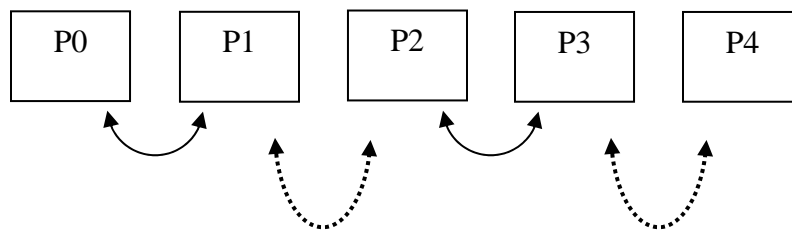


Figure 8. Esquema de la técnica usada

Después cada proceso envía la parte que le corresponde de la matriz A a los demás procesos para actualizar la matriz A en cada proceso.

Debemos de tener en cuenta que los vectores que describen la vida de cada cavidad y que guardan la matriz MA son de necesidad local en cada proceso ya que ellos solo son usados para guardar la información de las cavidades que luego será usada para producir la visualización.

3.7 Visualización

Realizar una imagen o una aproximación a la realidad es una buena idea para visualizar data o simulaciones. Sin embargo, estas imágenes pueden envolver muchos procesos y pueden ser altamente complejas, obscureciendo el análisis y dificultando la implementación, pero, en nuestro modelo la idea es lograr visualizar parámetros específicos en el transcurso del tiempo por lo tanto es necesario separarnos un poco de la realidad y ver la visualización como un método para entender mejor el comportamiento del modelo.

La visualización de nuestras simulaciones se da en dos partes, una parte que contiene los elementos esenciales para producir el mezclado (las cavidades) y otro donde podemos visualizar las dos fases que representan los dos fluidos.

3.7.1 Cavidades

En esta parte de la visualización comienza con la asignación a escala de los radios obtenidos en la ecuación 3.5 y además la asignación de una gama de colores para las energías.

Para representar las cavidades de forma visual utilizaremos esferas colocadas en cada punto del mallado, a dichas esferas se les asigna un radio a escala de los valores obtenidos de el vector r en la ecuación 3.5 (sección 3.3.4).

Además le asignamos a las superficies de las esferas un color que depende de una gama de colores que representan la energía de cada cavidad en un instante, esta asignación es de tal forma que a mayor energía mayor color rojo tendremos en la superficie de la esfera y menor energía mas azul es el color de la esfera

Con esta gama de colores adheridos a la superficie de cada esfera podemos ver con claridad los cambios de energía que ocurren en la cavidad como se puede ver en la figura 9.

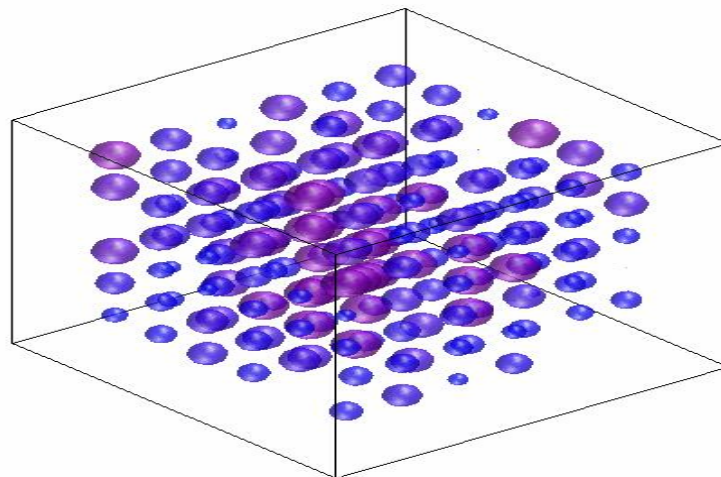


Figura 9. Gráfica de la visualización de las cavidades. (mallado de 6^3)

3.7.2 Mezcla (dos fases)

En la matriz de mezclado MA tenemos valores de 0 y 1, cada uno de ellos es para representar uno de los líquidos. Esta representación en la visualización es dada por capas o fases, cada una de un color como se ve en la grafica 10a. En el momento que tenemos un intercambio de masa, es decir un 1 pasa la posición de un 0, se crea una superficie compuesta por los elementos de un liquido inmerso en el otro como lo podemos ver en la figura 10b. Luego, uniendo estas dos graficas y usando transparencias en las fases, podemos tener una visualización en donde el mezclado es evidente, ver figura 10c.

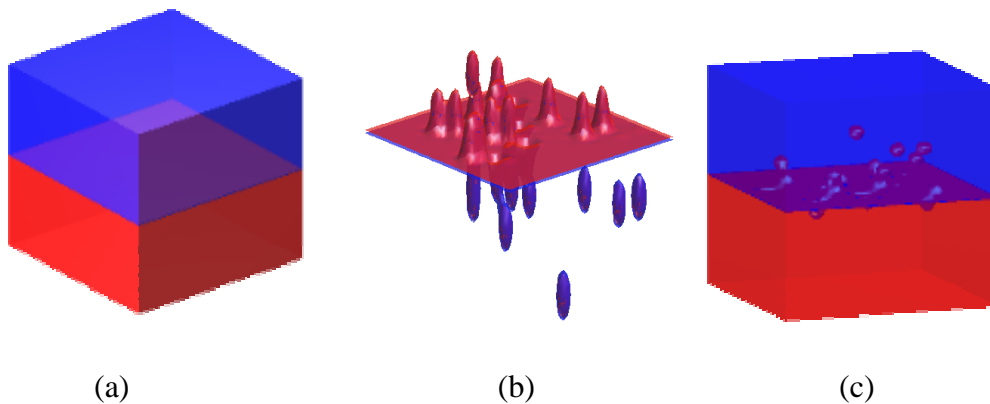


Figura 10. (a) Dos fases (b) Superficie generada por el intercambio de los fluidos (c) Unión de a y b

3.7.3 Un valor ó razón que sirva como medida del mezclado

Visualmente podemos presentar una aproximación del mezclado que se está realizando, pero esto no es suficiente, debemos desarrollar una idea clara y concisa de cómo medir el mezclado numéricamente y para esto realizamos los siguientes cálculos.

Para cada partícula en el volumen hacemos un conteo de todos los elementos que se encuentran alrededor, tomamos como referencia todos los que tienen color diferente a la partícula en cuestión y realizamos la siguiente razón $R_p = \frac{N_{ed}}{N_{te}}$ donde N_{ed} es el número de elementos diferentes de la partícula en cuestión y N_{te} es el número total de elementos que se encuentran alrededor de la partícula en cuestión, en la figura 11 Se puede ver que el mezclado es homogéneo si la cantidad $N_{ed}=14$ cuando $N_{te}=26$ con estos valores tenemos que un punto está mezclado si $R_p > 1/2$. Con R_p decidimos cuando un punto está mezclado, después obtenemos una razón de la siguiente forma $0 < \frac{\# \text{ puntos mezclado}}{\# \text{ puntos total}} < 1$, que dice cual es el porcentaje de mezclado.

Este método de medida es una aproximación bastante buena y que está directamente relacionada con la distribución de las partículas en cada instante de tiempo, de esta forma podemos calcular la razón en cada uno de los instantes y construir una curva de tiempo contra razón que nos de una buena descripción de cómo se comporta el mezclado

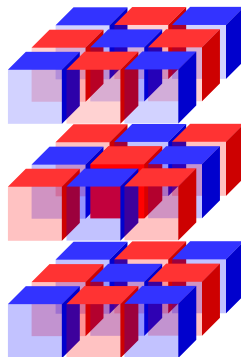


Figura 11. Partículas distribuidas en un mezclado homogéneo.

En la figura 12 se muestra un código de cómo se implementa esta medida. Esta función es ejecuta al final de cada instante de tiempo, cuando la matriz de mezcla MA está lista para la próxima iteración. Esta función solo usa como parámetro de entrada la matriz de mezcla MA y su resultado es un valor entre 0 y 1, es valor es interpretado de la siguiente forma, un valor mas cercano a uno (1) indica que el mezclado es bastante homogéneo y un valor cercano a cero (0) indica un mezclado pobre.

```
function vermT=verificam(MA)
vermT=0;
[m,n,l]=size(MA);
for i=2:n-1
    for j=2:m-1
        for k=2:l-1
            verm=0;
            for ii=-1:1
                for jj=-1:1
                    for kk=-1:1
                        if MA(i+ii,j+jj,k+kk)~=MA(i,j,k) & [ii jj kk]~=[0 0 0]
                            verm=verm+1;
                        end
                    end
                end
            end
            if verm/26 >1/2
                vermT=vermT+1;
            end
        end
    end
end
vermT=vermT/((n-2)*(m-2)*(l-2));
```

Figura 12. Función que encuentra una razón que mida el mezclado. (Código en Matlab)

4. RESULTADOS

4.1 Implementación en Matlab

El modelo descrito en el capítulo 3 fue primeramente implementado en MATLAB¹, y corrido en una máquina Pentium(R) 4 CPU 3.0GHz con 512 MB de RAM. El programa consta de dos partes: la primera se basó en la parte numérica del modelo y la segunda se basó en la visualización de los datos obtenidos en la parte uno. En las secciones siguientes se muestran los resultados.

4.1.1 Dos simulaciones

Las condiciones de los líquidos y de ambiente que se usaron se pueden observar en las Tablas 3, 4 y 5. Para la primera simulación, se usaron las condiciones dadas en las tablas 3 y 4, con estas condiciones se tiene un modelo que consta de un mallado de 6^3 cajas y un volumen que contiene 18^3 partículas, en la figura 13 se puede ver con claridad como está evolucionando el mezclado en dos tomas de 80 y 160 iteraciones. La figura 14 muestra el tiempo vs razón, lo cual representa el mezclado.

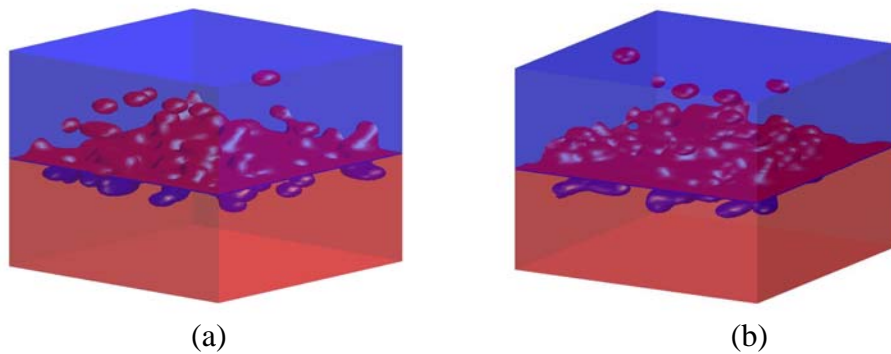
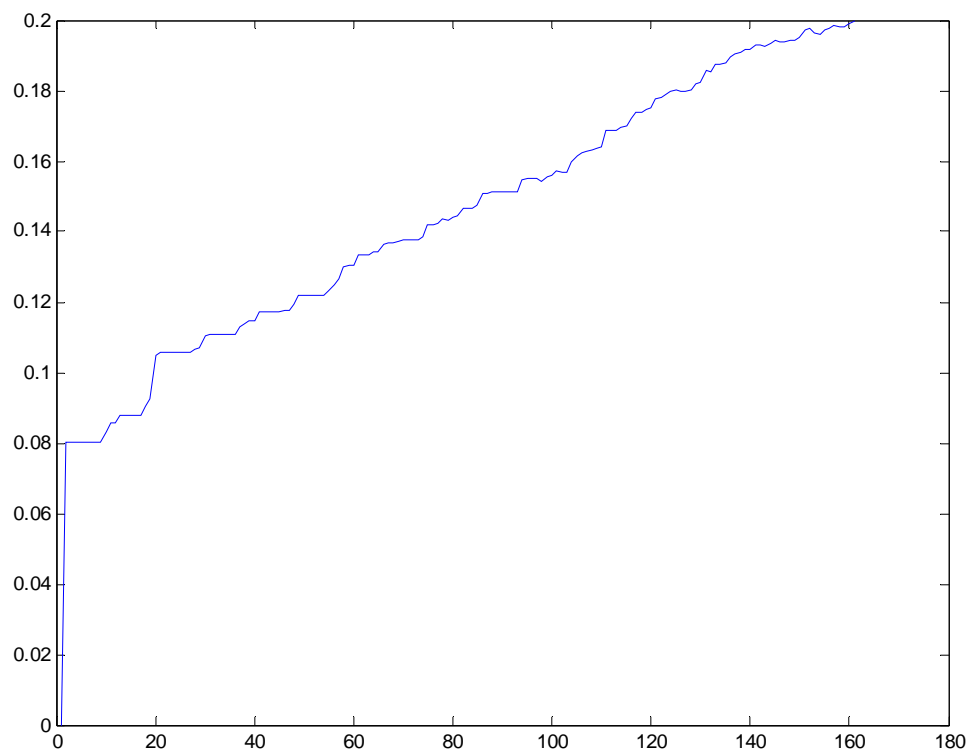
Tabla 3 .Constantes físicas.

| Densidad | viscosidad | Tensión superficial | Presión de vapor | Presión del ambiente |
|--------------------------|------------------|---------------------|------------------|----------------------|
| 899 Kg/m ³ | 0.001 Kg/m*s. | 0.0358 N/m | 2.00E+05 Pa | 1.13E+05 Pa |

¹ Matlab versión 6.5.0.1 release 13.

Tabla 4. Constantes para el modelo (primera simulación).

| Δt | Δx | Δs | L | Radio inicial | Frecuencia | Presión de amplitud |
|-------------|-------------|------------|-------------|---------------------------|--------------------------|---------------------------|
| 0.22-4 m | 1.2e-3 m | 4e-4 m | 0.0072 m | 2*10 ⁽⁻⁵⁾ m | Pi*10 ⁴ Hz | 2.4*10 ⁵ Hz |

**Figura 13.** Dos tomas del mezclado en tiempos (a) en la iteración 80 y (b) en la iteración 160.**Figura 14.** Instantes de tiempo vs razón numérica que representa el mezclado.

Para la segunda simulación, se usaron las condiciones dadas en las tablas 3 y 5. Con estas condiciones tenemos un modelo que consta de un mallado de 6^3 cajas y un volumen que contiene 18^3 partículas. En la figura 15 se puede observar con claridad como está evolucionando el mezclado en dos tomas de 80 y 160 iteraciones. Además en la figura 16 se puede ver una grafica de tiempo vs razón que representa el mezclado.

Tabla 5. Constantes para el modelo (segunda simulación).

| Δt | Δx | Δs | L | Radio inicial | Frecuencia | Presión de amplitud |
|---------------------------|--------------------------|------------------------|-------------|------------------------|-----------------------|------------------------|
| $0.22 \cdot 10^{-4}$ m | $1.2 \cdot 10^{-3}$ m | $4 \cdot 10^{-4}$ m | 0.0072 m | $2 \cdot 10^{-5}$ m | $10 \cdot 10^4$ Hz | $150 \cdot 10^5$ Hz |

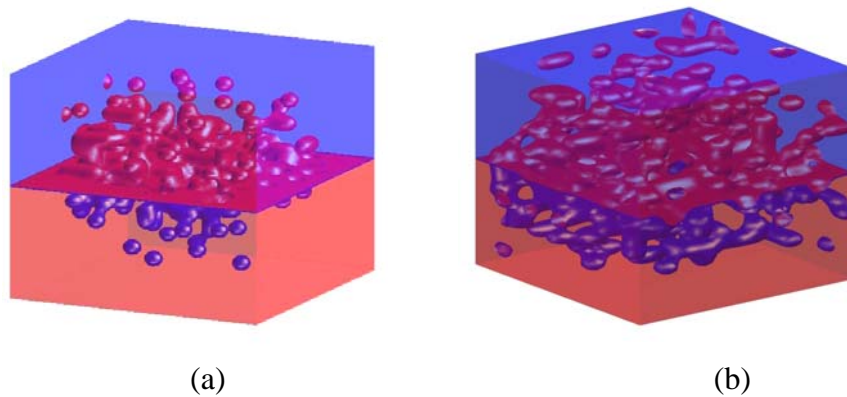


Figura 15. Dos tomas del mezclado en tiempos (a) en la iteración 80 y (b) en la iteración 160.

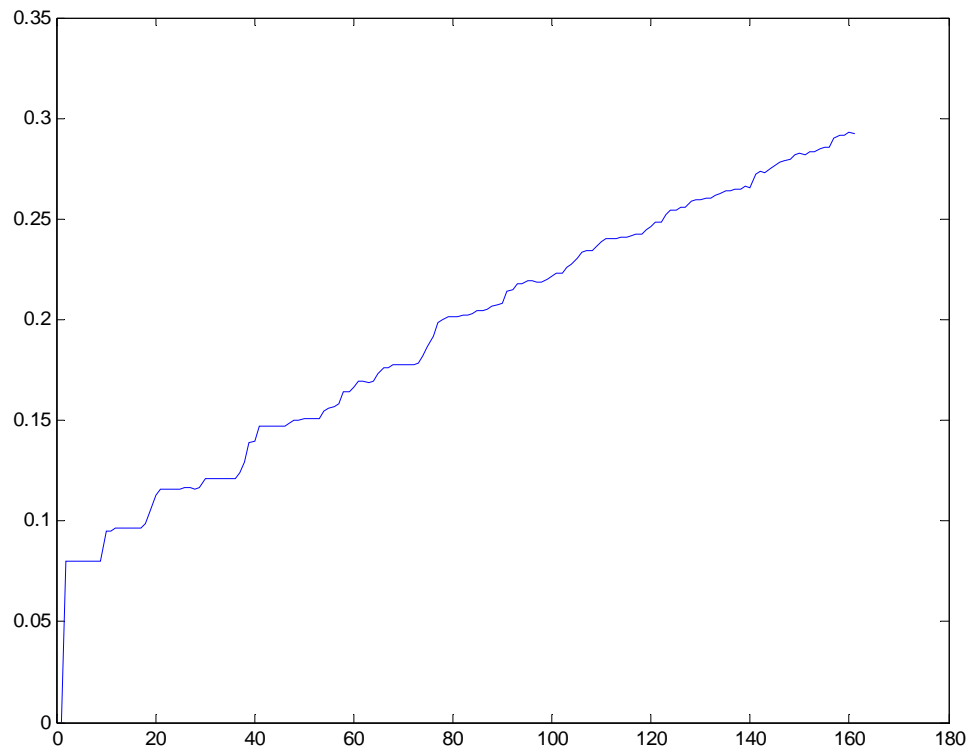


Figura 16. Instantes de tiempo vs razón numérica que representa el mezclado.

4.2 Rendimiento del Programa

A continuación se presentan algunos de los resultados obtenidos al ejecutar el programa en paralelo, en un sistema cluster con 4 procesadores SGI 230L 750MHz².

4.2.1 Speedup

La tabla 6 muestra los resultados obtenidos para hallar las semillas en matrices de tamaños 8, 12, 24, 36, 48, 60 y 72. Con esto se comprobó que la paralelización de esta

² Laboratorio de Visualización Científica (hpcf)

parte del programa es más eficiente cuando se hace en tamaños mayores de 24. Estos resultados se pueden ver en la grafica 17 en donde se cálculo el “Speedup”

$$Speedup = \frac{T_s}{T_p}$$

donde T_s es tiempo serial y T_p es tiempo en paralelo. Esto nos da una buena aproximación de lo que pasa con el mezclado ya que este es un cálculo que es esporádico y por lo tanto paralelizar este proceso causa aun menor eficiencia.

Table 6. Contiene el “*speedup*” de diferentes tamaños de la matriz A en el cálculo de la parte “semillas”.

| Tamaño\procesos | 1 | 2 | 3 | 4 |
|-----------------|---|----------|----------|----------|
| lineal | 1 | 2 | 3 | 4 |
| 8 | 1 | 0.197167 | 0.214715 | 0.114512 |
| 12 | 1 | 0.593845 | 0.553379 | 0.354311 |
| 24 | 1 | 1.354904 | 1.786539 | 1.989916 |
| 36 | 1 | 1.608657 | 2.262461 | 2.447628 |
| 48 | 1 | 1.696784 | 2.304436 | 2.726286 |
| 60 | 1 | 1.729514 | 2.364517 | 2.787819 |
| 72 | 1 | 1.754880 | 2.426440 | 2.891755 |

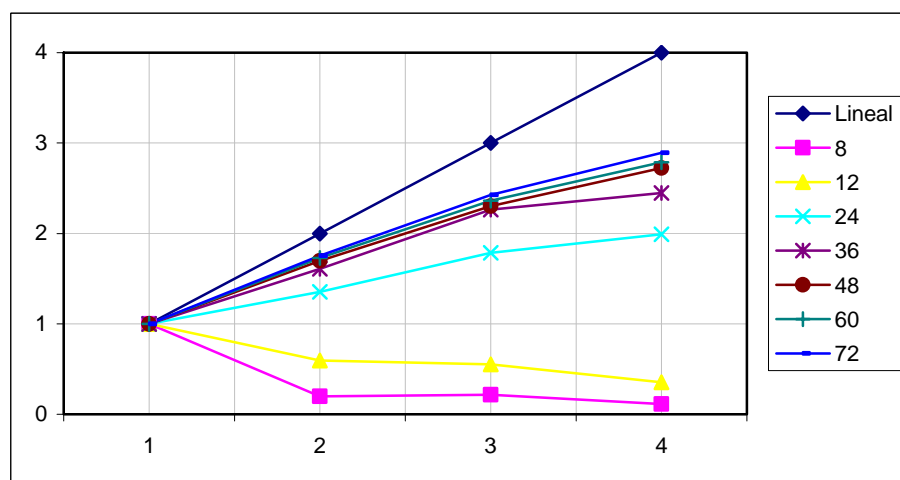


Figura 17. Gráfica del “*speedup*” Vs. aceleración lineal (semillas).

Los demás intervalos de tiempo

La tabla 7 muestra los tiempos obtenidos para hallar una simulación de 50 instantes de tiempo, en una matriz A de dimensión 20^3 con 4 procesadores. En él se pudo verificar la eficiencia del programa y además en la figura 18 vemos que no podemos asegurar que la escalabilidad sea buena.

Table 7. Resultados de ejecutar el programa con una matriz A de 20^3 .

| NP | 1 | 2 | 3 | 4 |
|-----------------------------|----------|----------|----------|----------|
| Tiempo | 369.3455 | 237.6327 | 160.3167 | 135.2401 |
| $Speedup = \frac{T_s}{T_p}$ | 1 | 1.55427 | 2.303849 | 2.731035 |
| $\frac{T_s}{pT_p}$ | 1 | 0.77714 | 0.76795 | 0.68276 |

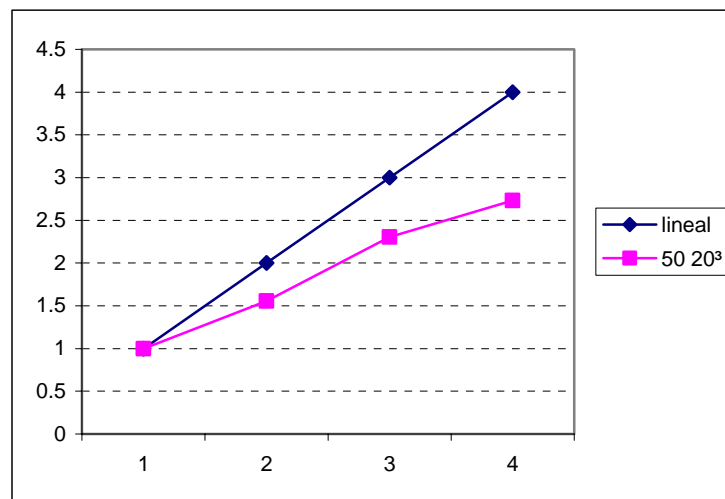


Figura 18. Gráfica del “*speedup*” Vs. aceleración lineal.
(50 iteraciones en un volumen de 20^3)

Las comunicaciones en este algoritmo paralelo juegan un factor importante al momento de explicar por puede estar sucediendo lo que vemos en la figura 18.

En la sección 3.6.3 podemos ver que en cada iteración (del `for` de tiempo) nosotros tenemos que hacer una actualización de la matriz **A** y para esto tenemos que realizar una cantidad de envíos que es de orden $O(N^3)$, esto puede estar causando la caída de la curva de “*speedup*”.

4.3 Implementación C, MPI y Matlab.

Por último tenemos una simulación cuya parte numérica es corrida en paralelo, en la tabla 8 se muestran los tiempos y el “*speedup*”. Los datos generados por esta simulación se guardan en archivo y luego de los archivos son cargados en MATLAB que es quien se encarga de realizar la visualización.

En la figura 19 se puede ver la escalabilidad del programa, y en la figura 20 y 21 se ven dos tomas de la visualización en la que se incluyen las dos partes (cavidades y mezclado) en que dividimos la visualización.

Tabla 8. Resultados de ejecutar el programa durante 160 iteraciones en una matriz A de 10^3

| NP | 1 | 2 | 3 | 4 |
|-----------------------------|------------|------------|------------|------------|
| Tiempo | 783.133071 | 432.360709 | 298.092638 | 254.153505 |
| $Speedup = \frac{T_S}{T_P}$ | 1 | 1.811296 | 2.627147 | 3.081339 |

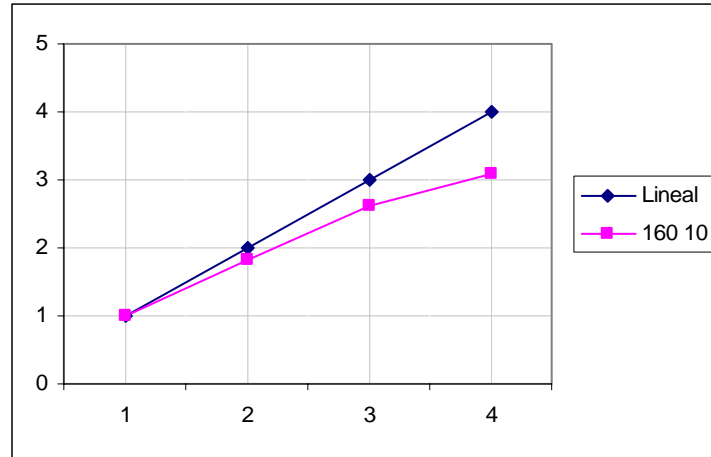


Figura 19. Gráfica del “*speedup*” Vs. aceleración lineal.
(160 iteraciones en un volumen de 10^3)

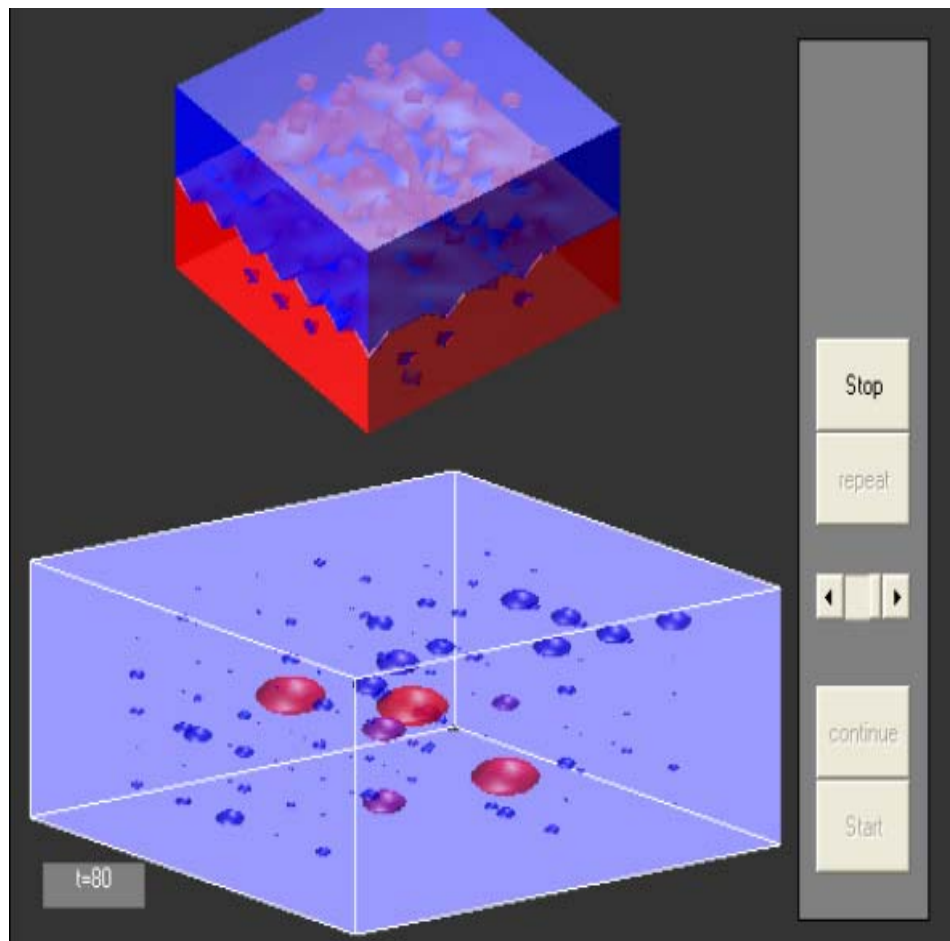


Figura 20. Simulación en la iteración 80, usando un mallado de 10^3 cajas y un volumen de 30^3 partículas

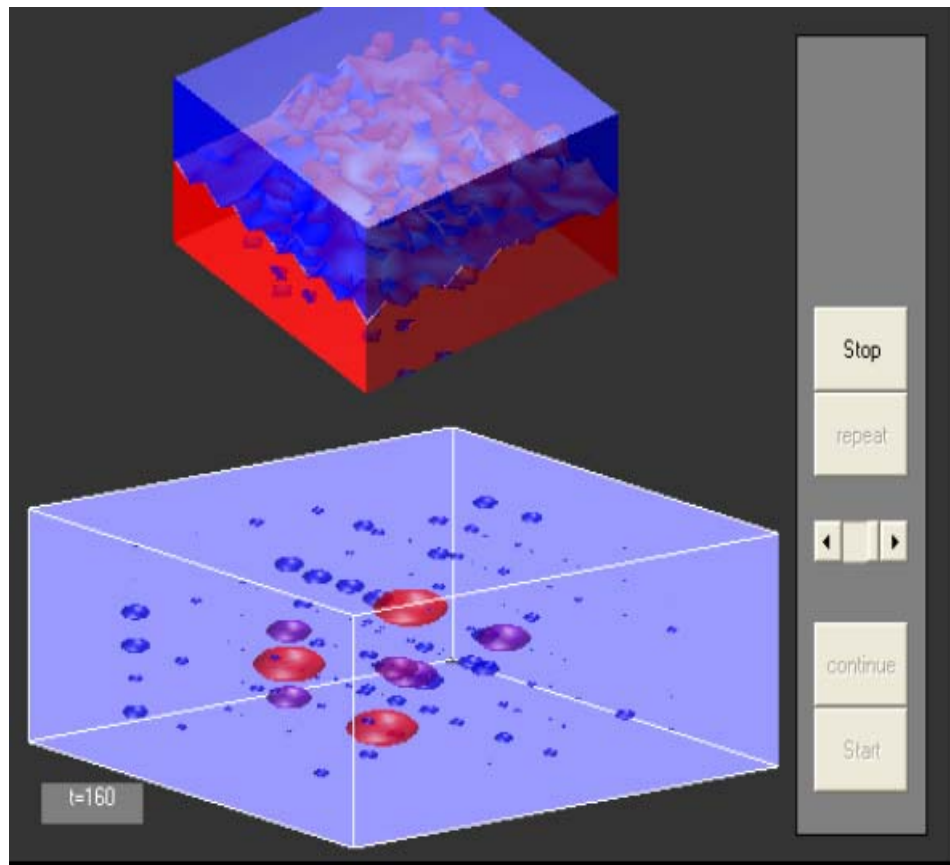


Figura 21. Simulación en la iteración 160, usando un mallado de 10^3 cajas y un volumen de 30^3 partículas

5. CONCLUSIONES

El objetivo de este estudio fue desarrollar un modelo matemático basado en heurísticas que permita simular el mezclado de líquidos inmiscibles mediante la generación de cavidades por estimulación ultrasónica.

Este modelo puede ser ajustado haciendo comparaciones con la realidad (en el laboratorio), esto lo podemos lograr cambiando o variando las heurísticas que estamos suponiendo, los resultados que obtuvimos hasta ahora no han sido comparados con datos de laboratorio pero parecen satisfactorios.

La utilización de MATLAB fue de gran importancia para generar cada una de las partes que componen el modelo y además nos sirvió como método de comparación a la hora de corroborar los datos obtenidos con los códigos en paralelo.

Las simulaciones realizadas están dentro de los rangos esperados ya que al usar los valores dados en las tablas 1 y 2 sección 4.1.1 se puede ver como cambia favorablemente el comportamiento del modelo. Con las gráficas de mezclado (figura 15) se pudo comprobar fácilmente que los valores que se sugieren en el estudio [3] tabla 5, son buenos parámetros para obtener una buena mezcla como los muestran la figura 16, donde se puede ver que la razón de mezclado alcanza un valor mayor en las condiciones de la tabla 4.

El uso de procesos locales y heurísticas es muy favorable ya que nos permite tener gran libertad al momento de implementar el modelo, esto ayuda a que los cálculos sean mucho menores y además nos permite tener una gran ventaja al momento de paralelizar.

En la sección 4.4 las tablas de tiempos y las graficas de “*speedup*” muestran que aunque no es óptimo paralelizar el proceso de creación de semilla en mallados pequeños, los demás procesos si aprovechan el paralelismo y el tiempo de cómputo se reduce de manera considerable para 2,3 y 4 procesadores. Es claro que la aceleración no está cerca de la aceleración lineal y por razones físicas no pudimos experimentar más arriba de 4 procesadores. Sin embargo basados en la información que tenemos parece que es muy posible que “*speedup*” siga decreciendo cuando la cantidad de procesadores crece.

La visualización usando Matlab, aunque es bastante fácil de implementar, no es lo más recomendable ya que este no es un paquete dedicado a la visualización, una idea futura seria implementar la visualización en un paquete especializado como lo es VTK[11].

Con los resultados obtenidos logramos “demostrar” que en realidad los “chorros” locales inducidos por las implosiones de cavidades son suficientes para generar un mezclado o emulsión muy fina, esto lo podemos concluir ya que en tan solo un corto tiempo de $t=0.00352$ se logro obtener un mezclado de aproximado del 30% en un volumen de 0.0072^3m^3 .

6 TRABAJOS FUTUROS

- En el modelo desarrollado, uno de las suposiciones mas simplistas que usamos es asumir que el reactor es infinito, una idea interesante seria trabajar con reactores finitos en donde las ondas de ultrasonido no se pierdan si no que regresen reflejadas en las paredes del reactor
- Implementar el algoritmo usando un modelo dinámico en el que se actualice el estado del sistema (con respecto a como cambia el ultrasonido que recibe una cavidad) en cada instante de tiempo.
- La visualización fue realizada usando MATLAB, por que es fácil de implementar, la visualización usando VTK seria una implementación más eficiente y podríamos tener volúmenes mucho más grandes donde se pueda simular un reactor de tamaño real.
- El método que usamos para el intercambio de masa es bastante simple, por tanto usar un modelo de colisión y propagación como lo es el método de lattice-Boltzman podría dar una descripción mas acertada de la mezcla.
- Experimentar más acerca de la relación entre T y el porcentaje de mezclado, para poder dar una aproximación del tiempo que hay que simular.
- El sistema donde se trabajó es bastante limitado en el número de procesadores, un buen trabajo estudiar el comportamiento de nuestro programa usando un cluster con muchos más procesadores.
- El estudio del algoritmo que desarrollamos nos permite decir que hacer paralelismo usando memoria compartida es una mejor opción para este modelo y para esto OpenMP seria la herramienta apropiada.

BIBLIOGRAFÍA

- [1] Buchanan & Turner. Numerical Methods and Analysis. McGraw Hill. 1992.
- [2] Burden & Faires. Análisis Numérico. Grupo Editorial Iberoamericana. 1985.
- [3] Carlos Javier Beltrán Sampayo. “Estudio del comportamiento de una burbuja dentro del proceso de producción de biodiesel mediante mezclado ultrasónico”. Thesis for the degree Master of Applied Mathematics. UPRM¹.
- [4] Ernesto Borrero Quintana. “Optimization studies for the alkaline transesterification Biodiesel reaction using ultrasound mixing”. Thesis for the degree Master of Science in Chemical Engineering. UPRM. 2002.
- [5] Fabio Alape. “ Effects of the use of ultrasonic waves over biodiesel production in alkaline transesterification of bleach tallow and vegetable oils : Cavitation Model. Thesis for the degree Master of Science in Chemical Engineering. UPRM.
- [6] Harkin, Nadim & Kaper. “On acoustic cavitation of slightly subcritical bubbles”. Physics of fluids. Volume II, number 2. (February 1999).
- [7] [Http://www.sonicsystems.co.uk](http://www.sonicsystems.co.uk)
- [8] P. J. Love, M. Nekovee, P. V. Coveney, J. Chin, N. González-Segredo, J. M. R. Martín, Simulations of amphiphilic fluids using mesoscale lattice-Boltzmann and lattice-gas methods.
- [9] Peter S. Pacheco, “Parallel Programming with MPI”, Morgan Kaufmann, 1997.
- [10] T.G. Leighton. The Acoustic Bubble. Academic Press. 1997.
- [11] Visualization ToolKit, Inc., VTK User's Guide.

APENDICE 1

Con base en el estudio [3] de la ecuación isotérmica y viscosa de Rayleigh-Plesset , haciendo el cambio de variables $x = R$ y $y = \dot{R}$ pasamos de la ecuación

$$\rho \left[R\ddot{R} + \frac{3}{2}\dot{R}^2 \right] + 4\mu \frac{\dot{R}}{R} = \left(p_0^\infty - p_v + \frac{2\sigma}{ri} \right) \left(\frac{ri}{R} \right)^3 + p_v - \frac{2\sigma}{R} - p_0^\infty + a \sin(ft) \quad (\text{A.1})$$

al siguiente sistema de ecuaciones no lineales para el cual $F : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ y $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ son funciones tales que si $X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$, se satisface el problema

$$X'(t) = F(x(t), y(t), t) \quad (\text{A.2})$$

o que es lo mismo

$$\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} y(t) \\ f(x(t), y(t), t) \end{pmatrix} = \begin{pmatrix} y \\ f \end{pmatrix} \quad (\text{A.3})$$

Donde

$$f(x, y, t) = -\frac{3y^2}{2x} - \frac{Ay}{x^2} + \frac{B}{x^4} + \frac{C}{x} - \frac{D}{x^2} + \frac{E \sin(ft)}{x} \quad (\text{A.4})$$

La función f es llamada rigid y la parte $E \sin(ft)$ es llamada estímulo. Las constantes A , B , C , D , E y F dependen de parámetros físicos establecidos que definiremos a continuación, dándoles también un rango de valores correspondientes a diferentes compuestos:

Tabla 9. Parámetros físicos con un rango de valores correspondientes a diferentes compuestos:

| | | |
|--|--|--|
| ρ | Densidad del liquido. | [732, 936] en Kg/m ³ |
| μ | Viscosidad. | [0.00035, 0.013] en Kg/m*s. |
| σ . | Surface tension of the liquid. | [0.0227, 0.0358] en N/m. |
| p_v | Presión de vapor. | [350, 35400] en Pa. |
| p_a | Pressure amplitude of ultrasonic field | . [10000, 110000] en Pa. |
| R_0 | Radio inicial. | [10 ⁻⁶ , 10 ⁻³] en m. |
| $A = \frac{4\mu}{\rho}$, | [1.5×10 ⁻⁶ , 7.1×10 ⁻⁵] | |
| $B = (p_v - p_a + \frac{2\sigma}{R_0}) \frac{R_0^3}{\rho}$ | [-0.12, 1.02×10 ⁻⁴] | |
| $C = \frac{p_a - p_v}{\rho}$ | [-27.137, 149.8] | |
| $D = \frac{2\sigma}{\rho}$ | [4.85×10 ⁻⁵ , 9.79×10 ⁻⁵] | |
| $E = \frac{p_a}{\rho}$ | [10.68, 150.28] | |
| $F = \Omega$ | [$\pi \times 10^3$, $\pi \times 10^6$]. | |

Para solucionar la ecuación necesitamos definir los parámetros anteriores que dependen en su gran mayoría del sistema y además definir como se aplica el factor de variación.

De la solución numérica de la ecuación anterior nosotros tomamos los vectores \bar{x} y \bar{y} que son el radio R y la velocidad del radio \dot{R} descrito numéricamente durante en el laxo de vida tv de la cavidad (tiempo de vida), estos vectores nos sirven para calcular el vector de energía mediante la formula dada en adelante

Energía dentro de la burbuja (4)

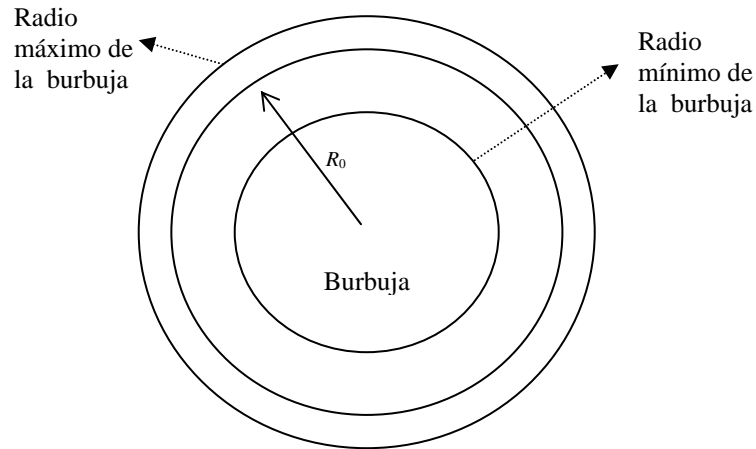


Figure 22. El radio de una burbuja, R , sometida a un estímulo ultrasónico varía entre un radio mínimo y un radio máximo; partiendo de un radio inicial R_0 .

de la fórmula 1, si $R = R(t)$ es el radio de la burbuja en un instante t , entonces la energía cinética dentro de la burbuja está dada por

$$\phi_k = \frac{1}{2} \rho \int_{R_0}^R \dot{r}^2 4\pi r^2 dr \quad (\text{A.5})$$

que por la condición de incompresibilidad nos da que

$$\phi_k = 2\pi\rho R^3 \dot{R}^2. \quad (\text{A.6})$$

por otro lado, el trabajo que se efectúa al interior de la burbuja para pasar de un volumen V_0 , que depende de R_0 , a un volumen V_R , que depende de R ; es la integral de $(p_g - p_0)dV$, esto es la energía potencial ϕ_p

$$\begin{aligned} \phi_p &= - \int_{R_0}^R (p_g - p_0) 4\pi r^2 dr \\ \phi_p &= (p_g - p_0) 4\pi (R^3 - R_0^3) \end{aligned} \quad (\text{A.7})$$

asumiendo que el gas se comporta politropicamente y que el proceso es isotermal, entonces $p_g V = \text{Constante}$; luego igualando la presión y volumen en condiciones de equilibrio da

$$\begin{aligned} p_g (R)^3 &= p_0 R_0^3 \\ \frac{p_g}{p_0} &= \left(\frac{R_0}{R} \right)^3 \end{aligned} \quad (\text{A.8})$$

de donde

$$p_g = p_0 \left(\frac{R_0}{R} \right)^3 \quad (\text{A.9})$$

por lo tanto, reemplazando (1.7) en (1.5), se obtiene que

$$\phi_P = - \left(p_0 \left(\frac{R_0}{R} \right)^3 - p_0 \right) 4\pi (R^3 - R_0^3) = \frac{4\pi p_0 (R^3 - R_0^3)^2}{R_0} \quad (\text{A.10})$$

finalmente, la energía total al interior de la burbuja, E_B , está dada por

$$E = \phi_K + \phi_P \quad (\text{A.11})$$

para con esto construir la función $(tv, \bar{\mathbf{R}}, \bar{\mathbf{E}}) = \mathbf{F}(ri, f, a, \bar{\mathbf{R}})$

Esta función describe el radio el tiempo discretamente nosotros usaremos un paso de tiempo lo suficiente mente conveniente para ver un descripción bastante aceptable del radio. Esto con el fin de tener un equilibrio entre descripción y tiempo de cómputo.

APENDICE 2

Programa De La Simulación En Forma Secuencial

```

#include <stdio.h>
#include <stdlib.h>
#include "libger.h"
#include "funcionopt.h"
#include <time.h>
#include <stdio.h>
#include <math.h>
#include <malloc.h>
#define PI 3.14159265358979323846

int main(int argc, char *argv[])
{

int
***A,***MAT,***MA,***B,***Ti,***Tv,N=6,Nup=3,dir[3],TIEMPO=30,i,li,ii,
j,jj,k,kk,t,ta,Numpar,xmax,ymax,zmax,Ra,T1,theta=15,tv;
float ***EL,***RL,
X[3],xHest[3],Dis,r,xi,xf,L,dx,dy,dz,P,U,Rp,Est[3],E,mp,masa;

int diff; /* Approximate time of a system call */
double ttime,TIMPI,TFMPI; /* Stores aprox. time spent in code segment */
struct timeval tv1, tv2; /* Structures for gettimeofday function */

r=0.3; /*probabilidad para la aprarion de cavidades*/
TIEMPO=atoi(argv[1]); /*iteraciones*/
N=atoi(argv[2]); /*tamaño de la matriz*/
A=cm3(N); /*matris de mayado de cavidades*/
MA=cm3(N+6); /*matris de mesclado temporal*/
MAT=cm4I(N+6); /*matris de mesclado (en el tiempo)*/
B=cm3(N); /*matris de mayado de cavidades temporal*/
Ti=cm3(N); /*matris de tiempos iniciales*/
Tv=cm3(N); /*matris de tiempos de vida*/
EL=cm4(N); /*matris de energías(en el tiempo) */
RL=cm4(N); /*matris de valores de radio (en el tiempo) */

dx=0.5; dy=0.5;dz=0.5;L=N/2;
Est[0]=L/(float)(2);Est[1]=L/(float)(2);Est[2]=L/(float)(2);
xmax=(int)(L/dx);
ymax=(int)(L/dy);
zmax=(int)(L/dz);
P=0.3;
T1=TIEMPO+25;
t=0;

gettimeofday(&tv1, NULL);

/** volumen inicial***/
for(i=0;i<N+6;i++)
for(j=0;j<N+6;j++)
for(k=0;k<N+6;k++)
{

```



```

        MAT[i][j][k]=(int *)malloc(T1*sizeof(int));
    if (k < N*r)
        MA[i][j][k]=1; /*representacion liquido uno*/
    else
        MA[i][j][k]=0; /*representacion liquido dos*/
        MAT[i][j][k][0]=MA[i][j][k]; /** Matris de mesclado en el
                                                tiempo cero*/
    }

    /**semillas**/
    for(i=0;i<N;i++)
        for(j=0;j<N;j++)
            {
                for(k=0;k<N;k++)
                    {
                        EL[i][j][k]=(float *)malloc(T1*sizeof(float));
                        RL[i][j][k]=(float *)malloc(T1*sizeof(float));
                        srand(time(0)+k+10*j+100*i);
                        Ra=rand();
                        U=(float)Ra/RAND_MAX;
                        if (U<P)
                            B[i][j][k]=1;
                    }
            }

    /**actualizando A a partir de B**/
    for(i=0;i<N;i++)
        for(j=0;j<N;j++)
            for(k=0;k<N;k++)
                A[i][j][k]=B[i][j][k];

    /**calculo de vida para las semillas iniciales*/
    for(i=0;i<N;i++)
        for(j=0;j<N;j++)
            for(k=0;k<N;k++)
                if (A[i][j][k]==1)
                    {
                        Rp=promedio(A,i,j,k,L,dx,theta);
                        Ti[i][j][k]=t;
                        tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,t);
                        Tv[i][j][k]=tv;
                    }

    /** calculo para los tiempos siguientes**/
    li=1;
    for(ta=1;ta<=TIEMPO;ta++)
        {

            for(i=0;i<N;i++)
                for(j=0;j<N;j++)
                    for(k=0;k<N;k++)
                        {
                            srand(time(0)+k+10*j+100*i);
                            if (A[i][j][k]==0)
                                {

```

```

        Rp=promedio(A,i,j,k,L,dx,theta);
        X[0]=i*dx;X[1]=j*dy;X[2]=k*dz;
        xHest[0]=X[0]-Est[0];
        xHest[1]=X[1]-Est[1];
        xHest[2]=X[2]-Est[2];
        Dis=norma(xHest);
        P=0.3;
Ra=rand();
        U=(float)Ra/(float)(RAND_MAX);
        if (U<P)
            {
                B[i][j][k]=1;
                Ti[i][j][k]=ta;
                tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,ta);
                Tv[i][j][k]=tv;
            }
        /*si la cavidad existía pero murió se realiza
                                                un mezclado en ese lugar */
        else
            {
                if (Ti[i][j][k]+Tv[i][j][k]==ta)
                    {
                        /*direccion aleatoria */
                        srand(time(0)+k+10*j+100*i);
                        dir[0]=((3*rand())/RAND_MAX)-1;
                        srand(time(0)+k+10*j+100*i);
                        dir[1]=((3*rand())/RAND_MAX)-1;
                        srand(time(0)+k+10*j+100*i);
                        dir[2]=((3*rand())/RAND_MAX)-1;
                        /*cantidad de patriculas*/
                        E=EL[i][j][k][ta];
                        mp=899*(3e-4*3e-4*3e-4);
                        masa=2*E/(1467*1467);
                        Numpar=masa/mp;
                        mezcla(MA,i+Nup,j+Nup,k+Nup,Numpar,&dir[0]);
                        B[i][j][k]=0;
                    }
            }
        }

        /**actualizando A y guardando el mezclado**/

        for(ii=0;ii<N+2*Nup;ii++)
            for(jj=0;jj<N+2*Nup;jj++)
                for(kk=0;kk<N+2*Nup;kk++)
                    {
                        if (ii<N & jj<N & kk<N )
                            A[ii][jj][kk]=B[ii][jj][kk];
                            MAT[ii][jj][kk][ta]=MA[ii][jj][kk];
                    }
    }
}

```

```

gettimeofday(&tv2, NULL);
ttime = tv2.tv_sec - tv1.tv_sec;
ttime = ttime + (tv2.tv_usec - tv1.tv_usec) / 1.0e6;
printf(" tiempo    %lf\n",ttime);
return 0;

}

```

Programa De La Simulación En Forma Paralela

```

#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>
#include "libger.h"
#include "funcionopt.h"
#include <time.h>
#include <math.h>
#include <malloc.h>
#include <sys/types.h>
#include <unistd.h>
#define PI 3.14159265358979323846

int main(int argc, char *argv[])
{

int size,rank,len;
int v1,v2,r,pi,pd,pr;
int
***A,***MAT,***MA,***B,***C,***Ti,***Tv,N,Nup=3,Numpar,IN,FN,TIEMPO=16
0,dir[3],i,ii,j,jj,k,kk,t,ta,xmax,ymax,zmax,f,Ra,Tl,theta=15,tv;
int VEI[3][100],VED[3][100],VER[3][100];
FILE *fichero;
float ***EL,***RL, ***T,
X[3],xHest[3],Dis,xi,xf,dx,L,t1,t2,dy,dz,P,U,Rp,Est[3],b,E;
double mp,masa;

int diff;
double ttime,TIMPI,TFMPI;

MPI_Init(&argc, &argv);
/*char name[MPI_MAX_PROCESSOR_NAME];*/
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
MPI_Comm_size(MPI_COMM_WORLD,&size);
/*MPI_Get_processor_name(name,&len);*/

r=0.3;/*probabilidad para la aprarion de cavidades*/
TIEMPO=atoi(argv[1]);/*iteraciones*/
N=atoi(argv[2]); /*tamaño de la matriz*/
A=cm3(N); /*matris de mayado de cavidades*/
MAT=cm4I(N+6); /*matris de mesclado (en el tiempo)*/
MA=cm3(N+6); /*matris de mesclado temporal*/

```

```

B=cm3(N);           /*matris de mayado de cavidades temporal*/
Ti=cm3(N);         /*matris de tiempos iniciales*/
Tv=cm3(N);         /*matris de tiempos de vida*/
EL=cm4(N);         /*matris de valores de energias (en el tiempo) */
RL=cm4(N);         /*matris de valores de radio (en el tiempo) */
f=26000;
dx=0.5; dy=0.5;dz=0.5;L=N/2;
Est[0]=L/(float)(2);Est[1]=L/(float)(2);Est[2]=L/(float)(2);
xmax=(int)(L/dx);
ymax=(int)(L/dy);
zmax=(int)(L/dz);
P=0.3;
T1=TIEMPO+25;
t=0;

/**particion del volumen a lo largo de un eje**/

TIMPI=MPI_Wtime();

for(i=0;i<size;i++)
  if(rank==i)
  {
    IN=(i*N)/size;FN=((i+1)*N)/size;
    printf("    %d  %d  %d %d valores de tope \n",IN,FN,rank,size );
  }

  /**semillas y volumen inicial***/
for(i=IN;i<FN;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
      {
        if (k < N*r)
          MA[i][j][k]=1; /*representacion del liquido uno*/
        else
          MA[i][j][k]=0; /*representacion de liquido dos*/
        EL[i][j][k]=(float *)malloc(T1*sizeof(float));
        RL[i][j][k]=(float *)malloc(T1*sizeof(float));
        srand(time(0)+k+10*j+100*i);
        Ra=rand();
        U=(float)Ra/RAND_MAX;
        if (U<P)
          {
            A[i][j][k]=1;
            B[i][j][k]=1;
          }
      }

  /**actualizando A, B y MA para todos lo procesos*/

for(k=0;k<size;k++)
  {
    for(i=(k*N)/size;i<((k+1)*N)/size;i++)
      for(j=0;j<N;j++)

```



```

        P=0.3;
        /*P=1-Dis*Rp/(1+Dis);*/
        Ra=rand();
        U=(float)Ra/(float)(RAND_MAX);
        /*si aparece la cavidad se calcula la vida de la cavidad*/
        if (U<P)
        {
            B[i][j][k]=1;
            Ti[i][j][k]=ta;
            tv=rayle(&RL[i][j][k][0],&EL[i][j][k][0],1-Rp,ta);
            Tv[i][j][k]=tv;
        }
    }
/*si la cavidad existia pero murio se realiza un mezclado en ese
lugar*/
else
{
    if (Ti[i][j][k]+Tv[i][j][k]==ta)
    {
        /*direccion aleatoria del mezclado*/
        srand(time(0)+k+10*j+100*i);
        dir[0]=((3*rand())/(RAND_MAX))-1;
        srand(time(0)+k+10*j+100*i);
        dir[1]=((3*rand())/(RAND_MAX))-1;
        srand(time(0)+k+10*j+100*i);
        dir[2]=((3*rand())/(RAND_MAX))-1;

        E=EL[i][j][k][ta];
        mp=899*(3.3e-4*3.3e-4*3.3e-4);
        masa=2*E/(1467*1467);
        Numpar=masa/mp;
        /*si el punto es de frontera save en vector de calculos de frontera*/
        if (i<=IN+Nup && rank!=0)
        {
            VEI[0][pi]=i;
            VEI[1][pi]=j;
            VEI[2][pi]=k;
            VEI[3][pi]=Numpar;
            pi++;
        }
        else
        if (i>=FN-Nup && rank!=size-1)
        {
            VED[0][pd]=i;
            VED[1][pd]=j;
            VED[2][pd]=k;
            VED[3][pd]=Numpar;
            pd++;
        }
        /* si no es de frontera mezclar*/
        else
        {
            mezcla(MA,i+3,j+3,k+3,Numpar,&dir[0]);
            B[i][j][k]=0;
        }
    }
}

```

```

    }
  }
}

/*tranferencia de informacion entre procesos de dos en dos*/

/*envio de un proceso par a uno impar */
if(rank==0 && rank!=size-1)
{
  for(i=IN;i<FN;i++)
  for(j=0;j<N;j++)
  {
    MPI_Ssend(&MA[i][j],N+6,MPI_INT, 1,0,MPI_COMM_WORLD);
  }
  MPI_Ssend(&pd,1,MPI_INT, rank+1,0,MPI_COMM_WORLD);
  for(i=0;i<3;i++)
    MPI_Ssend(&VED[i],pd,MPI_INT, rank+1,i,MPI_COMM_WORLD);
}
else
{
  k=rank-1;
  for(i=(k*N)/size;i<((k+1)*N)/size;i++)
  for(j=0;j<N;j++)
  {
    /*printf(" recibel  %d  %d  %d  rank
%d\n",k,i,j,rank );*/

    MPI_Recv(&MA[i][j],N+6,MPI_INT,0,0,MPI_COMM_WORLD,MPI_STATUS_IGNORE)
;
  }

  MPI_Recv(&pr,1,MPI_INT,rank-
1,0,MPI_COMM_WORLD,MPI_STATUS_IGNORE);
  for(i=0;i<3;i++)
    MPI_Recv(&VER[i],pr,MPI_INT,rank-
1,i,MPI_COMM_WORLD,MPI_STATUS_IGNORE);

  /*calculos de frontera*/

  for(i=0;i<pr;i++)
  {
mezcla(MA,VER[0][i],VER[1][i],VER[2][i],VER[3][i],&dir[0]);
    B[VER[0][i]][VER[1][i]][VER[2][i]]=0;
  }
  for(i=0;i<pr;i++)
  {
mezcla(MA,VER[0][i],VER[1][i],VER[2][i],VER[3][i],&dir[0]);
    B[VER[0][i]][VER[1][i]][VER[2][i]]=0;
  }
}

```

```

    }
    /*renvio de informacion*/
    if(rank%2==0 && rank!=size-1)
        for(i=IN;i<FN;i++)
            for(j=0;j<N;j++)
                {

MPI_Recv(&MA[i][j],N,MPI_INT,rank+1,i,MPI_COMM_WORLD,MPI_STATUS_IGNO
RE);
                }
            else
            {
k=rank-1;
for(i=(k*N)/size;i<((k+1)*N)/size;i++)
    for(j=0;j<N;j++)
        {

            MPI_Ssend(&MA[i][j],N,MPI_INT, rank-1,i,MPI_COMM_WORLD);
        }

    }

/*envio de un proceso impar a uno par */

if(rank%2!=0 && rank!=size-1)
{
    for(i=IN;i<FN;i++)
        for(j=0;j<N;j++)
            {
                printf(" envial 0  %d  %d  rank  %d\n",i,j,rank );
                MPI_Ssend(&MA[i][j],N,MPI_INT, rank+1,i,MPI_COMM_WORLD);
            }
        MPI_Ssend(&pd,1,MPI_INT, rank+1,0,MPI_COMM_WORLD);
        for(i=0;i<3;i++)
            MPI_Ssend(&VED[i],pd,MPI_INT, rank+1,i,MPI_COMM_WORLD);

    }
    else
    {
k=rank-1;
for(i=(k*N)/size;i<((k+1)*N)/size;i++)
    for(j=0;j<N;j++)
        {

            MPI_Recv(&MA[i][j],N,MPI_INT,rank-
1,i,MPI_COMM_WORLD,MPI_STATUS_IGNORE);
        }

        MPI_Recv(&pr,1,MPI_INT,rank-
1,0,MPI_COMM_WORLD,MPI_STATUS_IGNORE);
    }

```



```

        for(i=0;i<3;i++)
            MPI_Recv(&VER[i],pr,MPI_INT,rank-
1,i,MPI_COMM_WORLD,MPI_STATUS_IGNORE);

        /*calculos de frontera*/
        for(i=0;i<pr;i++)
        {

mezcla(MA,VER[0][i],VER[1][i],VER[2][i],VER[3][i],&dir[0]);
            B[VER[0][i]][VER[1][i]][VER[2][i]]=0;
        }
        for(i=0;i<pr;i++)
        {

mezcla(MA,VER[0][i],VER[1][i],VER[2][i],VER[3][i],&dir[0]);
            B[VER[0][i]][VER[1][i]][VER[2][i]]=0;
        }

    }

    /*renvio de informacion*/
    if(rank%2==0 && rank!=size-1)
        for(i=IN;i<FN;i++)
            for(j=0;j<N;j++)
            {
                printf(" envial 0  %d  %d  rank  %d\n",i,j,rank );

                MPI_Recv(&MA[i][j],N,MPI_INT,rank+1,i,MPI_COMM_WORLD,MPI_STATUS_IGNO
RE);
            }
        else
        {
            k=rank-1;
            for(i=(k*N)/size;i<((k+1)*N)/size;i++)
                for(j=0;j<N;j++)
                {
                    /*printf(" recibel  %d  %d  %d  rank
%d\n",k,i,j,rank );*/
                    MPI_Ssend(&MA[i][j],N,MPI_INT, rank-1,i,MPI_COMM_WORLD);
                }
        }

}

/**actualizando A para todos lo procesos*/

for(k=0;k<size;k++)
    for(i=(k*N)/size;i<((k+1)*N)/size;i++)
        for(j=0;j<N;j++)
        {
            MPI_Bcast(&B[i][j], N, MPI_INT,k , MPI_COMM_WORLD);
        }

```

```
/** SAVE Matris de mesclado en el tiempo ta*/
    if(rank==0)
        for(ii=0;ii<N+6;ii++)
            for(jj=0;jj<N+6;jj++)
                for(kk=0;kk<N+6;kk++)
                    MAT[ii][jj][kk][ta]=MA[ii][jj][kk];

/* actualizando A apartir de B*/
    for(ii=0;ii<N;ii++)
        for(jj=0;jj<N;jj++)
            for(kk=0;kk<N;kk++)
                {
                    A[ii][jj][kk]=B[ii][jj][kk];
                }

    }

    TFMPI=MPI_Wtime();
    ttime=TFMPI-TIMPI;
    printf(" tiempo %d    %lf \n",rank,ttime);

MPI_Finalize();
return 0;

}
```

LIBRERÍA DE FUNCIONES

funcionopt.h

```

#include<stdlib.h>
#include<math.h>
#include<stdio.h>
#include <malloc.h>
#define PI 3.14159265358979323846

void mezcla(int*** M,int x,int y,int z,int lon,int dir[])
{
    int i,r1i,r2i,r3i,r1f,r2f,r3f,aux;
    for(i=0;i<lon;i++)
    {
        r1i=dir[0]*i;
        r2i=dir[1]*i;
        r3i=dir[2]*i;
        r1f=dir[0]*(i-lon);
        r2f=dir[1]*(i-lon);
        r3f=dir[2]*(i-lon);
        aux=M[x+r1i][y+r2i][z+r3i];
        M[x+r1i][y+r2i][z+r3i]=M[x+r1f][y+r2f][z+r3f];
        M[x+r1f][y+r2f][z+r3f]=aux;
    }
}

long double stimulus(long double E,long double F,long double t)
{
    long double stimulu=E*sin(F*t);
    return stimulu;
}

float energia(float ri,float den,float X1,float X2,float PA,float PAM)
{
    long double Pg;
    float R3, RP2, Ecinetica, Epotencial, Energia;
    R3=X1*X1*X1;
    RP2=X2*X2;
    Ecinetica=2*PI*R3*den*RP2;
    Pg=PAM*(ri/X1)*(ri/X1)*(ri/X1);
    Epotencial=abs((-4)*(PI/3)*(Pg-PAM)*(R3-ri*ri*ri));
    Energia=Ecinetica+Epotencial;
    return Energia;
}

void rigid(long double dy[],long double h,long double t,long double
y1,long double y2,long double A,long double B,long double C,long double
D,long double E,long double F)
{

```

```

    dy[0]=h*y2;
    dy[1]=((double)(-3)/(double)(2))*(y2*y2)/y1-A*y2/(y1*y1)+
    B/(y1*y1*y1*y1)+C/y1-D/(y1*y1)+stimulus(E,F,t)/y1;
    dy[1]=h*dy[1];

}

float norma(float g[])
{
    float r;
    r=sqrt(g[0]*g[0]+g[1]*g[1]+g[2]*g[2]);
    return r;
}

float promedio(int*** A,int x,int y,int z,float L,float dx,int theta)
{
    int i,j,k,ii,xmax,red;
    float Rp,g[3],sum=0,count=0,rg,rx,X[3],GX,gX,gHx[3],cosangle,E;

    xmax=(int)(L/dx);
    g[0]=x*dx-L/2;
    g[1]=y*dx-L/2;
    g[2]=z*dx-L/2;
    rg=norma(g);
    cosangle=cos(theta*PI/180);

    for(i=0;i<xmax;i++)
        for(j=0;j<xmax;j++)
            for(k=0;k<xmax;k++)
                {
                    X[0]=(float)i*dx-L/2;
                    X[1]=(float)j*dx-L/2;
                    X[2]=(float)k*dx-L/2;
                    rx=norma(X);
                    gHx[0]=g[0]-X[0];
                    gHx[1]=g[1]-X[1];
                    gHx[2]=g[2]-X[2];
                    gX=g[0]*X[0]+g[1]*X[1]+g[2]*X[2];
                    GX=(float)(gX)/(rg*rx);
                    if (GX>cosangle && GX<=(rg*rg) && norma(gHx)<=rg )
                        {
                            sum=sum+A[i][j][k];
                            count=count+1;
                        }
                }
            if (count==0 || sum==0)
                Rp=0;
            else
                Rp=(float)(sum/count);
            if (Rp>=0.9)
                Rp=0.9;

    return Rp;
}

int rayle(float X1s[],float X2s[],float Rp,int ti)

```

```

{
  int tv,i,u,j,tan,k;
  double tiem, delpos;
  long double b, YA[2], y[2], w[2], K1[2], K2[2], K3[2],
  K4[2],Kw1[2],Kw2[2],Kw3[2],Kw4[2];
  float *T, X1[4000], X2[4000];

  long double densidad=899, viscosidad=0.001,
  tensionsup=0.0358,presionvap=2e5;
  long double presionamb=1.1325e5, difpres,Dn[2];
  long double A,B,C,D,E,razondh;
  long double radioest=2e-6, PA, F=1e5;
  long double hmax, h,h2,t,ta, hmin, tol, tolmin;
  difpres=presionamb-presionvap;
  PA=(double)Rp*150e5;

  A=4*viscosidad/densidad;
  B=(-difpres+2*tensionsup/radioest)*(pow(radioest,3))/densidad;
  C=difpres/densidad;
  D=2*tensionsup/densidad;
  E=PA/densidad;

  h=0.0000023949126;
  tol=0.00001;
  tolmin=tol/(double)32;

  t=0;
  y[0]=radioest;
  y[1]=0;
  X1[0]=y[0];
  X2[0]=y[1];
  w[0]=radioest;
  w[1]=0;

  i=0;
  while ( fabs(y[1])<=1460 )
  {
    YA[0]=y[0];
    YA[1]=y[1];
    w[0]=YA[0];
    w[1]=YA[1];

    rigid(&K1[0],h,t,y[0],y[1],A,B,C,D,E,F);

    rigid(&K2[0],h,t+h/(double)(2),y[0]+(double)(1)/(double)(2)*K1[0],y[1]+
    (double)(1)/(double)(2)*K1[1],A,B,C,D,E,F);

    rigid(&K3[0],h,t+h/(double)(2),y[0]+(double)(1)/(double)(2)*K2[0],y[1]+
    (double)(1)/(double)(2)*K2[1],A,B,C,D,E,F);
    rigid(&K4[0],h,t+h,y[0]+K3[0],y[1]+K3[1],A,B,C,D,E,F);

    y[0]=y[0]+(double)(1)/(double)(6)*(K1[0]+(double)(2)*K2[0]+(double)(2)*
    K3[0]+K4[0]);

    y[1]=y[1]+(double)(1)/(double)(6)*(K1[1]+(double)(2)*K2[1]+(double)(2)*
    K3[1]+K4[1]);
  }
}

```

```

    ta=t;
    h2=(double)(h)/(double)(2);
    for (u=1;u<=2;u++)
    {

        rigid(&Kw1[0],h2,ta,w[0],w[1],A,B,C,D,E,F);

    rigid(&Kw2[0],h2,ta+h2/(double)(2),w[0]+(double)(1)/(double)(2)*Kw1[0],
    w[1]+(double)(1)/(double)(2)*Kw1[1],A,B,C,D,E,F);

    rigid(&Kw3[0],h2,ta+h2/(double)(2),w[0]+(double)(1)/(double)(2)*Kw2[0],
    w[1]+(double)(1)/(double)(2)*Kw2[1],A,B,C,D,E,F);
        rigid(&Kw4[0],h2,ta+h,w[0]+Kw3[0],w[1]+Kw3[1],A,B,C,D,E,F);

    w[0]=w[0]+(double)(1)/(double)(6)*(Kw1[0]+(double)(2)*Kw2[0]+(double)(2)
    )*Kw3[0]+Kw4[0]);

    w[1]=w[1]+(double)(1)/(double)(6)*(Kw1[1]+(double)(2)*Kw2[1]+(double)(2)
    )*Kw3[1]+Kw4[1]);
        ta= ta + h2;
    }

    Dn[0]=(w[0]-y[0])/(double)(15);
    Dn[1]=(w[1]-y[1])/(double)(15);

    razondh=Dn[0]/h;
    j++;
    if (tol < fabs(razondh))
    {
        h=h/(double)(2);
        y[0]=YA[0];
        y[1]=YA[1];
        w[0]=YA[0];
        w[1]=YA[1];
    }
    if (tolmin <= fabs(razondh) && fabs(razondh) <= tol)
    {
        y[0]=w[0];
        y[1]=w[1];
        t= t + h;
        i=i+1;
        X1[i]=y[0];
        X2[i]=y[1];
    }
    if (fabs(razondh) < tolmin)
    {
        t= t + h;
        i=i+1;
        X1[i]=y[0];
        X2[i]=y[1];
        h=(double)(2)*h;
    }

}
}
tv=i-1;

```

```
tiem=t/0.00001;
k=(int)(tiem);
delpos=(double)(tv)/tiem;

X1s[ti+0]=X1[0];
X2s[ti+0]=energia(radioest,densidad,X1[0],X2[0],PA,presionamb);

for (u=1;u<k;u++)
{
    X1s[ti+u]=X1[(int)(u*delpos)];

X2s[ti+u]=energia(radioest,densidad,X1[(int)(u*delpos)],X2[(int)(u*delpos)],PA,presionamb);
}

X1s[ti+k]=X1[tv];
X2s[ti+k]=energia(radioest,densidad,X1[tv],X2[tv],PA,presionamb);

/* printf("%d \n",tv); */

return k;
}
```

libger.h

```

#include<stdlib.h>

int*** cm3(int f)
{
    int ***g = NULL;
    int i,j;
    g=(int ***)malloc(f*sizeof(int **));

    for(j=0;j<f;j++)
        g[j]= (int **) malloc(f*sizeof(int *));
    for(j=0;j<f;j++)
        for(i=0;i<f;i++)
            g[j][i]=(int *)malloc(f*sizeof(int));
    return g;
}

float**** cm4(int f1)
{
    float ****g=NULL;
    int i,j;
    g=(float ****)malloc(f1*sizeof(float ***));

    for(j=0;j<f1;j++)
        g[j]=(float ***)malloc(f1*sizeof(float **));
    for(j=0;j<f1;j++)
        for(i=0;i<f1;i++)
            g[j][i]=(float **)malloc(f1*sizeof(float *));
    return g;
}

int**** cm4I(int f1)
{
    int ****g=NULL;
    int i,j;
    g=(int ****)malloc(f1*sizeof(int ***));

    for(j=0;j<f1;j++)
        g[j]=(int ***)malloc(f1*sizeof(int **));
    for(j=0;j<f1;j++)
        for(i=0;i<f1;i++)
            g[j][i]=(int **)malloc(f1*sizeof(int *));
    return g;
}

void dm(float **g,int f)
{
    int i;
    for(i=0;i<f;i++)
        { free(g[i]);
          g[i]=NULL;
        }
    free(g);
    g=NULL;
}

```