



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

CENTRO DE INVESTIGACIÓN EN INGENIERÍA Y CIENCIAS APLICADAS

“Optimización de la convergencia en la transferencia de calor, mediante el uso de una heurística computacional”

TESIS PARA OBTENER EL GRADO DE:

**DOCTOR EN INGENIERÍA Y CIENCIAS APLICADAS CON OPCIÓN TERMINAL
EN TECNOLOGÍA MECÁNICA**

PRESENTA:

M.C. JUANA ENRÍQUEZ URBANO

**DIRECTORES: DR. MARCO ANTONIO CRUZ CHÁVEZ Y DR. JESÚS XAMÁN
VILLASEÑOR**

**SINODALES: DR. SERGIO ALONSO SERNA BARQUERA, DR. JUAN CARLOS
GARCÍA CASTREJÓN, DRA. JESÚS DEL CARMEN PERALTA ABARCA, DR.
MARTÍN HERIBERTO CRUZ ROSALES, DRA. BEATRIZ MARTÍNEZ BAHENA.**

CUERNAVACA MORELOS, DICIEMBRE 2019



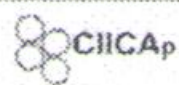
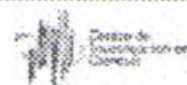
UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Instituto de
Investigación en
Ciencias
Básicas y
Aplicadas

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

Defensoría de Posgrado en Ingeniería y Ciencias Aplicadas



"1919-2019: en memoria del General Emiliano Zapata Salazar"

Cuernavaca, Morelos, a 08 de noviembre de 2019.

DR. ROSENBERG JAVIER ROMERO DOMÍNGUEZ
COORDINADOR DEL POSGRADO EN
INGENIERÍA Y CIENCIAS APLICADAS
P R E S E N T E

Atendiendo a la solicitud para emitir DICTAMEN sobre la revisión de la TESIS "Optimización de la convergencia en la transferencia de calor, mediante el uso de una heurística computacional" que presenta la alumna **JUANA ENRÍQUEZ URBANO**, para obtener el título de **DOCTORADO EN INGENIERÍA Y CIENCIAS APLICADAS** con opción terminal en **TECNOLOGÍA MECÁNICA**.

Nos permitimos informarle que nuestro voto es:

NOMBRE	DICTAMEN	FIRMA
DR. SERGIO ALONSO SERNA BARQUERA	Aprobado	
DR. JUAN CARLOS GARCÍA CASTREJÓN	Aprobado	
DRA. JESÚS DEL CARMEN PERALTA ABARCA (FCQel)	Aprobado	
DR. MARTÍN HERIBERTO CRUZ ROSALES, (FCAel)	Aprobado	
DRA. BEATRIZ MARTÍNEZ BAHENA (FCQel)	Aprobado	
DR. JESÚS XAMÁN VILLASEÑOR (CENIDET)	Aprobado	
DR. MARCO ANTONIO CRUZ CHÁVEZ	Aprobado	

PLAZO PARA LA REVISIÓN 20 DÍAS HÁBILES (A PARTIR DE LA FECHA DE RECEPCIÓN DEL DOCUMENTO)

NOTA. POR CUESTION DE REGLAMENTACIÓN LE SOLICITAMOS NO EXCEDER EL PLAZO SEÑALADO, DE LO CONTRARIO LE AGRADECEMOS SU ATENCIÓN Y NUESTRA INVITACIÓN SERÁ CANCELADA.

RESUMEN

El diseño de técnicas y algoritmos eficientes que resuelvan adecuadamente problemas complejos de optimización es uno de los campos dentro de la investigación en el área de la ingeniería, dar una solución adecuada y eficiente a este tipo de problemas cobra cada vez mayor importancia tanto para la industria como para la comunidad científica. Además, que, tanto los requisitos temporales como computacionales son un factor crítico a tener en cuenta cuando se trata de resolver problemas de grandes proporciones cuya solución debe ser aplicable en un supuesto real. En estos casos, el uso de heurísticas y metaheurísticas se presenta como una alternativa de solución viable en un tiempo razonable, por otro lado, la eficiencia de dichos algoritmos heurísticos como los resultados que éstos obtienen, mejoran de manera significativa cuando se aplican estrategias de paralelismo. Este es uno de los contextos en los que se desarrolla esta tesis doctoral.

En este trabajo se aborda una propuesta de optimización de la convergencia del tiempo computacional para un problema de dinámica de fluidos mediante la aplicación de la meta heurística de recocido simulado (SA), la cual obtiene factores de relajación que optimiza la convergencia durante la ejecución numérica del problema. Los resultados experimentales muestran que para el problema de transferencia de calor se mejora la convergencia en tiempo computacional hasta un 70%.

Una de las aplicaciones que permite este trabajo es encontrar mejores diseños en equipos o edificios aplicados al confort térmico. Acelerar la solución de los problemas de transferencia de calor, permite conocer más rápidamente diferentes configuraciones en fenómenos para la transferencia de calor y elegir la mejor para aplicarla al confort térmico.

ABSTRAC

The design of efficient techniques and algorithms that solve complex optimization problems is one of the fields within the research in the area of engineering, to give an adequate and efficient solution to these types of problems is becoming increasingly important for both the industry and the scientific community. In addition to the fact that both the temporal and computational requirements are a critical factor to consider when it comes to solving large-scale problems whose solution must be applicable in a real assumption. In these cases, the use of heuristics and metaheuristics is presented as a viable solution alternative in a reasonable time, on the other hand the efficiency of said heuristic algorithms as the results they require, improve significantly when it comes to parallelism strategies. Which is one of the contexts in which this doctoral thesis is developed.

This paper addresses a proposal to optimize the convergence of computational time for a fluid dynamics problem by applying the heuristic simulated annealing (SA) goal, which obtains relaxation factors that optimize convergence during numerical execution of the problem. Experimental results show that convergence in computational time is improved up to 70% for the heat transfer problem.

One of the applications that allows this work is to find better designs in equipment or buildings applied to thermal comfort. Accelerate the solution of heat transfer problems, allows to know more quickly different configurations in phenomena for heat transfer and choose, the best one to apply it to thermal comfort

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología por el apoyo económico que me otorgó para la realización de mis estudios de maestría.

Instituto de Investigación en Ciencias Básicas y Aplicadas (IICBA) por permitirme realizar mis estudios y brindarme lo suficiente para que concluyera satisfactoriamente.

A mis asesores el Dr. Marco Antonio Cruz y al Dr. Jesús P. Xáman que con sus comentarios y consejos, me ha apoyado a lograr un objetivo más en mi vida.

A los miembros del jurado revisor: Dr. Sergio Alonso Serna, Dr. Juan Carlos García, Dra. Jesús del Carmen Peralta, Dr. Martín Heriberto Cruz, Dra. Beatriz Martínez, que con su tiempo, interés y comentarios, han enriquecido este trabajo de tesis.

A todos mis profesores por ser partícipes en mi formación profesional, por sus consejos, por su tiempo y paciencia.

A mi esposo, amigo y compañero Daniel, que compartió de cerca cada momento del doctorado, el cual ha sido es y siempre será un pilar en mi vida.

A mis amigos y amigas de CIICAp: Alfonso, Dra. Carmen, Betty, Pedrito, Alina, Jaz Ari, Mtra. Marta y Yainier por las buenas experiencias que nos unieron como compañeros.

Y a todos los que de alguna u otra manera colaboraron con éste trabajo..... Muchas gracias!

Dedicatorias

*Ante todo a **Dios** por darme el regalo de la vida, unos padres ejemplares, una familia y amigos que me ha apoyado en los momentos que más lo he necesitado.*

*A mi mamá **Milburga Urbano (EPD)** que con su ejemplo amor y cariño me ayudó a superar cada una de las etapas de mi vida en las que estuvo presente, a mi papá **Pedro Enríquez** que de la mano de mi mamá me inculcaron los buenos valores, el respeto por la vida y con su vida son un ejemplo para mi de servicio y perseverancia.*

*A mí querida familia: mi esposo **Jacinto D. Frías**, a mis hijos **Daniela, Benedicto, Mateo y Miriam**. Que siempre me han apoyado y juntos hemos vencido las adversidades que día a día se presentan.*

*A mis hermanos **Miriam y Heriberto** que a lo largo de las diferentes etapas de mi vida han estado presentes y son partícipes de mis éxitos y mis alegrías, y apoyo cuando lo he necesitado y es por eso que sé que siempre voy a contar con ustedes y ustedes conmigo.*

*A mi nueva familia **Lili, Martha, P. Benedicto, Karina**, y todos los **Frías** que me han apoyado incondicionalmente en estos últimos años de mi vida y que de alguna manera cada uno de ellos ha contribuido para lograr este objetivo.*

A mis amigos y a todos aquellos que me han apoyado y aconsejado para no rendirme ante las dificultades.

CONTENIDO

CONTENIDO	i
ÍNDICE DE TABLAS	iv
ÍNDICE DE FIGURAS	vi
NOMENCLATURA	viii
Capítulo 1. INTRODUCCIÓN	
1.1 Motivación	2
1.2 Optimización combinatoria	4
1.2.1 Heurísticas	5
1.3 Antecedentes (estado del arte)	6
1.4 Objetivo General	9
1.4.1 Objetivos específicos	9
1.5 Contribución de la tesis	9
1.6 Organización de la tesis	10
Capítulo 2. MARCO TEÓRICO	
2.1 Descripción de problema (casos de prueba)	12
2.1.1 Introducción	12
2.1.2 Caso 1: Problema convección-conducción	12
2.1.3 Modelo matemático para el problema convección- conducción	13
2.1.4 Caso 2: Problema hidrodinámico	14
2.1.5 Modelo matemático del problema hidrodinámico	15
2.2 Metodología de la solución	16
2.2.1 Generación de la malla computacional	16

2.2.2	Concepto de malla desplazada	17
2.2.3	Discretización del modelo matemático	18
2.2.3.1	Esquemas de interpolación de bajo orden	19
2.2.3.2	Esquema centrado	19
2.2.3.3	Esquema Upwind (corrientes arriba)	22
2.2.3.4	Esquema Híbrido	23
2.2.3.5	Esquema Ley de Potencia	25
2.2.4	Algoritmo SIMPLE	29
2.2.4.1	Gradiente de Presión	30
2.2.4.2	Formulación del algoritmo SIMPLE	31
2.2.4.3	Relajación de la solución parcial	43
Capítulo 3.	METODOLOGÍA DE LA SOLUCIÓN	
3.1	Metodología de la solución aplicada al algoritmo de recocido simulado	47
3.1.1	Introducción	47
3.1.2	Técnica metaheurística SA	49
3.2	Función de costo para optimizar los factores de relajación	61
3.3	Estrategia para optimizar los factores de relajación	62
3.3.1	Estructura de vecindad para recocido simulado	64
3.3.2	Algoritmo de recocido simulado (secuencial)	65
3.4	Paralelización de algoritmos	68
3.4.1	Técnicas de paralelización	68
3.4.2	Message Passing Interface (MPI)	70
3.4.3	Modelos de comunicación entre procesos	73
3.4.4	Tipos de comunicación	75
3.4.5	Recocido simulado (paralelo)	77
3.5	Análisis y sintonización de parámetros	79

Capítulo 4.	PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS	
4.1	Resultados experimentales para el algoritmo secuencial	81
4.2	Resultados experimentales para el algoritmo en paralelo	94
4.2.1	Infraestructura del Clúster Cuexcomate	95
4.2.2	Resultados experimentales	97
Capítulo 5.	PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS	
5.1	Conclusiones	99
5.2	Trabajos futuros	100

ÍNDICE DE TABLAS

Tabla 1.1	Clases de Complejidad para la toma de decisión de problemas (Talbi E., 2009)	5
Tabla 2.1	Propiedades físicas del problema hidrodinámico	15
Tabla 2.2	Tabla 2.2 Condiciones iniciales del problema hidrodinámico	15
Tabla 2.3	Función $A(P_e)$, con respecto al esquema numérico	28
Tabla 3.1	Analogía entre proceso de recocido simulado en metalurgia, recocido simulado en optimización combinatoria y los casos de prueba	52
Tabla 4.1	Valores sintonizados para los parámetros de control de recocido simulado	81
Tabla 4.2	Ejecuciones realizadas por SA para una malla de 41x41 y un residual de $\varepsilon = 1x10^{-05}$, para el problema del caso 1.	83
Tabla 4.3	Factores de relajación obtenidos con SA, con mallas y residual relajados ($\varepsilon = 1x10^{-05}$), para el problema del caso 1.	85
Tabla 4.4	Factores de relajación obtenidos con SA, con malla y residual ($\varepsilon = 1x10^{-10}$) no relajados, para el problema del caso 1.	86
Tabla 4.5	Tabla 4.5. Valores sintonizados para los parámetros de control de recocido simulado para el caso de prueba 2.	86
Tabla 4.6	Ejecuciones realizadas para una malla de 11x11 un residual de $\varepsilon = 1x10^{-05}$, para el problema del caso 2.	87
Tabla 4.7	Factores de relajación obtenidos con SA, con diferentes tamaños de mallas y residual de ($\varepsilon = 1x10^{-05}$) para el problema del caso 2	90

Tabla 4.7	Factores de relajación obtenidos con SA, con malla de 61x61 y residual ($\varepsilon = 1x10^{-10}$) no relajados, para el problema del caso 2	93
Tabla 4.9	Infraestructura Hardware y Software del Clúster Cuexcomate, Peralta 2016	96
Tabla 4.10	Distribución de los recursos del Clúster Cuexcomate	97
Tabla 4.11	Distribución de procesos de nodos Cuexcomate	97
Tabla 4.12	Tiempos de ejecución del algoritmo distribuido del problema caso 2, para una malla de 41x41 y un $\varepsilon = 1x10^{-5}$	98

ÍNDICE DE FIGURAS

Figura 2.1	Modelo físico para el problema de convección-conducción de calor 2D, (J. Xamán et al. 2016)	13
Figura 2.2	Modelo físico para el problema hidrodinámico, (J. Xamán et al. 2016)	14
Figura 2.3	Representación de volúmenes de control en 2-D.	17
Figura 2.4	Representación de mallas superpuestas: a) VC para variables escalares, b) VC para u_e y c) VC para v_n , (J. Xamán et al. 2016)	18
Figura 2.5	Volumen de control sujeto a análisis.	20
Figura 2.6	Variación de a_E con el número de Peclet (P_e), (J. Xamán et al. 2016)	24
Figura 2.7	Desviación del esquema híbrido respecto al valor exacto de a_E/D_e , (J. Xamán et al. 2016).	26
Figura 2.8	Función a_E/D_e para el esquema ley de potencia, (J. Xamán et al. 2016).	27
Figura 2.9	Malla principal (●), malla desplazada en dirección - x () y malla desplazada en dirección - y (Δ), (J. Xamán et al., 2016).	32
Figura 2.10	Volumen de control (VC) frontera para la ecuación de continuidad (J. Xamán et al., 2016).	42
Figura 3.1	Ejemplo de un mínimo local	47
Figura 3.2	Clasificación de algunas de las técnicas de aproximación	49
Figura 2.8	Configuraciones de un sólido aplicando recocido simulado	50
Figura 3.4	Pseudocódigo del algoritmo de recocido simulado	56

Figura 3.5	Representación de un espacio de soluciones aplicando el algoritmo de recocido simulado	60
Figura 3.6	Recocido simulado implementado al problema de convección-conducción y al problema hidrodinámico	68
Figura 3.7	Estructura básica de un algoritmo en MPI	73
Figura 3.8	Modelo Maestro-Eslavo	74
Figura 3.9	Modelo Cliente-Servidor	75
Figura 3.10	Comunicación punto a punto	76
Figura 3.11	Comunicación colectiva	77
Figura 3.12	Momentos de comunicación entre maestro y esclavos	78
Figura 3.13	Diagrama del algoritmo de recocido simulado distribuido	79
Figura 4.1	Análisis de sensibilidad para los parámetros de SA	82
Figura 4.2	Comportamiento en el tiempo de convergencia del problema del caso 1 en función del factor de relajación	84
Figura 4.3	Algoritmo SA. Optimización del tiempo de convergencia del problema hidrodinámico, caso 2. Malla= 11x11, $\varepsilon = 1 \times 10^{-05}$	85
Figura 4.4	Factores de relajación para una malla de 11x11 y un residual de $\varepsilon = 1 \times 10^{-05}$, para el problema del caso 2	89
Figura 4.5	Comparación del tiempo computacional (t) con respecto al N° de iteraciones para la solución del problema hidrodinámico para diferentes factores de relajación (los propuestos y los encontrados con SA)	91
Figura 4.6	Tiempo de convergencia del problema hidrodinámico con SA	92
Figura 4.7	Componentes de velocidad dentro de la cavidad	94
Figura 4.8	Clúster Cuexcomate	95

NOMENCLATURA

Símbolos

$\alpha_P, \alpha_E, \alpha_W, \alpha_N, \alpha_S$	<i>Coefficientes de la ecuación discretizada</i>
$A(Pe)$	<i>Función de Peclet</i>
b	<i>Término fuente</i>
C_p	<i>Calor específico</i>
dx, dy	<i>Diferencia en dirección $-x, y$</i>
D_e, D_w, D_n, D_s	<i>Conductancia o flux difusivo en la interface del VC</i>
dt	<i>Diferencial del tiempo</i>
F_e, F_w, F_n, F_s	<i>Flux convectivo</i>
$f(s)$	<i>Función de costo</i>
$Frelax$	<i>Factor de relajación</i>
$Frelax_T$	<i>Factor de relajación para la temperatura</i>
$Frelax_{u,v,P}$	<i>Factor de relajación en dirección u, v, y el de la presión (P)</i>
H, H_x, H_y	<i>Dimensión o longitud del sistema</i>
J_e, J_w, J_n, J_s	<i>Flux total en la interface del VC</i>
P	<i>Presión</i>
Pe	<i>Número de Peclet</i>
Re	<i>Número de Reynolds</i>
s	<i>Espacio de soluciones</i>
S_{actual}	<i>Solución actual</i>
S_0	<i>Solución inicial</i>
S_{vecina}	<i>Solución vecina</i>
T	<i>Temperatura</i>
t	<i>Tiempo</i>
t_{conv}	<i>Tiempo de convergencia</i>
$T_{SA,0}$	<i>Parámetro de control inicial de SA</i>
$T_{SA,f}$	<i>Parámetro de control final de SA (criterio de paro)</i>
u, v	<i>Componentes de velocidad en dirección $-x, y$</i>
V	<i>Volumen</i>
x y y	<i>Coordenadas en dirección $-x, y$</i>

Símbolos griegos

α	<i>Parámetro de control que determina la velocidad de enfriamiento</i>
Γ	<i>Coefficiente de transporte difusivo</i>
$\delta x, \delta y$	<i>Distancia entre nodos computacionales en dirección $-x, y$</i>

Δt	<i>Espesor del VC temporal</i>
$\Delta x, \Delta y$	<i>Espesor del VC espacial en dirección $-x, y$</i>
ΔE	<i>Probabilidad de aceptación de Boltzmann</i>
λ	<i>Conductividad térmica del aire</i>
μ	<i>Viscosidad dinámica del aire</i>
ρ	<i>Densidad</i>
\emptyset	<i>Variable dependiente general ($u, v, P, T, etc.$)</i>

Subíndices

<i>conv</i>	<i>Convergencia</i>
<i>e, E</i>	<i>Este</i>
<i>n, N</i>	<i>Norte</i>
<i>s, S</i>	<i>Sur</i>
<i>w, W</i>	<i>Oeste</i>
<i>k</i>	<i>Contador</i>
<i>P, E, W, N, S</i>	<i>Puntos de la malla</i>

Siglas

CFD	<i>Computational Fluid Dynamics</i>
DISS	<i>Direct Solar Steam</i>
EDP	<i>Ecuaciones Diferenciales Parciales</i>
EDPs	<i>Ecuaciones diferenciales parciales</i>
ETR	<i>Ecuación de Transferencia Radiativa</i>
GMRES	<i>Generalized Minimal Residual algorithm</i>
HO	<i>Alto-orden</i>
HR	<i>Alta-resolución</i>
LCM	<i>Longitud de la Cadena de Markov</i>
MPI	<i>Message Passing Interface</i>
MVF	<i>Método de volúmenes finito</i>
NWF	<i>Weighting-factor normalizado</i>
RS	<i>Recocido Simulado</i>
SA	<i>Simulated Annealing</i>
SIMPLE	<i>Método Semi-implícito para ecuaciones acopladas a la presión.</i>
SIMD	<i>Single Instruction Multi Data</i>
SPMD	<i>Single Program Multi Data</i>
VC	<i>Volumen de Control</i>

CAPITULO I INTRODUCCIÓN

1.1 Motivación

La simulación computacional es una herramienta poderosa para el análisis y comprensión de diversos fenómenos físicos que ocurren en la naturaleza y en los procesos industriales, los cuales son objetos de estudios en los diferentes campos de investigación. En el campo de la investigación de la ingeniería en especial la mecánica para el desarrollo de simulaciones computacionales es necesario resolver ecuaciones diferenciales parciales (EDP), utilizando diversos métodos numéricos como por ejemplo Diferencias Finitas, Elementos Finitos y Volúmenes Finitos. Estos métodos transforman (Discretizan) las EDP de un dominio finito en un sistema de ecuaciones algebraicas en un dominio discreto, la solución de este sistema de ecuaciones resulta una etapa de elevado tiempo computacional.

Generalmente, la solución numérica de los problemas de flujos de fluidos, transferencia de calor y de masa requiere de un proceso iterativo debido principalmente a la no-linealidad del problema, para detener o terminar el proceso iterativo de manera automática en un número k de iteraciones, se requiere de un criterio de paro o mejor conocido como criterio de convergencia. La solución final obtenida en la iteración k debe satisfacer las ecuaciones discretizadas del problema. El criterio de convergencia debe ser establecido de tal manera que la diferencia entre la solución de la iteración actual k y la solución de la iteración anterior ($k-1$), debe ser mínima. Esta diferencia durante el proceso iterativo debe decrecer. En otras palabras, la solución numérica debe ser convergente.

Por otro lado, cuando se usa un método iterativo para la solución del sistema de ecuaciones algebraicas resultantes de método numérico de discretización, durante el proceso iterativo se puede acelerar o alentar los cambios de la variable ϕ entre una iteración y la siguiente. Este artificio numérico es llamado sobre-relajación (*overrelaxation*) o bajo-relajación (*underrelaxation*), dependiendo si los cambios de la variable es acelerada o disminuida, respectivamente. La sobre-relajación es comúnmente usado en conjunto con el método de Gauss-Seidel y el esquema

resultante es conocido como “Sucesiva sobre-relajación” (*Successive Over-Relaxation*, SOR).

Si se utilizan valores para el factor de relajación (α) en el intervalo $0 < \alpha < 1$, el efecto es de baja-relajación. Por otro lado, si el valor de α es mayor que 1 se produce una sobre-relajación.

El valor óptimo de α depende del caso bajo estudio a resolver. Un valor de $\alpha \approx 1$ permite acelerar el proceso iterativo y lo hace sensible a situaciones de divergencia; por el contrario, un valor de α cercano a cero retarda el proceso iterativo y lo ayuda a evitar una divergencia. Aunque en general, es necesario emplear la experiencia y la intuición para elegir el valor más apropiado de α para cada caso, existen pocos estudios de investigación donde se ha empleado algún tipo de teoría para optimizar el valor de α durante el proceso iterativo.

En el desarrollo de esta tesis se aborda este tipo de problemas mediante la aplicación de una heurística computacional para obtener los valores cercanos a los óptimos de los factores de relajación, durante la solución numérica de problemas de transferencia de calor por convección. Se considera un problema de transferencia de calor en una cavidad es 2D con flujo por convección natural; es elegido este tipo de problema como caso de prueba principalmente por la no linealidad que se presenta en el desarrollo del modelo matemático que lo representa, lo cual puede llegar a presentar inestabilidad al momento de darle solución numérica. Sin embargo, existen problemas de transferencia de calor en cavidades en 2D con régimen turbulento por convección mixta, y forzada, donde principalmente en la convección mixta se presentan patrones de flujo bastante inestables durante su solución numérica y para llegar a la convergencia se requiere amplia experiencia en la modelación de estos sistemas de lo contrario podría no obtenerse una solución; o bien usar factores de relajación con valores bajos, lo cual llevaría a tiempos de computo excesivos

1.2 Optimización combinatoria

Se puede definir el concepto de optimización en las áreas de ingeniería e informática como el método para determinar los valores de las variables que intervienen en un proceso o sistema para que el resultado sea el mejor posible.

La optimización combinatoria es una rama de la optimización cuyos problemas se centran en problemas de decisión. Problemas tales como la minimización en el uso de recursos empresariales, la ruta más corta en la distribución de productos y el ruteo de vehículos en empresas de transporte por mencionar algunos que forman parte de este tipo de problemas.

*Un problema de optimización viene dado por un par (F, c) donde F es un dominio de puntos factibles y c una función de costo. La problemática consiste en encontrar un punto factible $x \in F$ tal que $\forall y \in F$ verifique $c(x) \leq c(y)$. Cada punto de x verificando las condiciones anteriormente descritas se denomina **óptimo global** del problema.*

Todos los problemas considerados en la literatura por la Optimización Combinatoria se pueden clasificar de acuerdo al grado de dificultad para resolverlos, de aquí surge la *Teoría de la Complejidad*, la cual clasifica a dicho problemas como problemas P, NP, NP-Completo. Los problemas P, son problemas de decisión los cuales pueden ser acotados por un algoritmo determinístico con prueba de optimalidad en tiempo polinomial, es decir, que la relación entre el tamaño del problema y su tiempo de ejecución es polinómica. Por ende, se dice que los problemas P son relativamente “fáciles” de resolver. Los problemas NP, son problemas de decisión que pueden ser acotados por un algoritmo no determinístico en tiempo polinomial. Una característica de estos problemas, es que su espacio de soluciones es de comportamiento exponencial conforme se incrementa el tamaño de la instancia (Martínez-Oropeza 2010). Para los problemas NP-Completo, son los problemas más duros dentro del conjunto NP, es decir, estos son aún más difíciles de resolver, ver Figura 1.

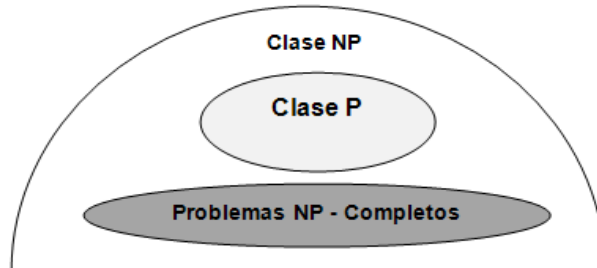


Figura 1.1. Clases de Complejidad para la toma de decisión de problemas (Talbi E., 2009)

Si bien, el problema planteado para el cálculo de los factores de relajación no se conoce su grado de dificultad (puesto que no es la problemática que se está planteando), de acuerdo a la clasificación que hace la *Teoría de la Complejidad* se puede resolver por medio de un algoritmo no determinístico como es el caso de las técnicas heurísticas; como se puede observar en la Figura 1, aun cuando fuera considerado como un problema P esta clasificación se encuentra dentro de la clasificación de problemas NP por lo que se pueden utilizar los mismos métodos de solución, adecuándolos para cada problema en particular. Suponiendo que el problema de factores de relajación fuera NP si se tratara de resolver con una técnica clásica exacta, el tiempo para resolver el problema tendría un comportamiento exponencial, no así si se tratara de resolver con una heurística computacional dado que el tiempo para encontrar una solución aproximada del problema se acota en tiempo polinomial.

1.2.1 Heurísticas

Existen problemas que presentan un alto grado combinatorio, es decir, es difícil generar un modelo basado en programación matemática que represente exactamente una situación real. A lo largo de la historia se han diseñado diversos métodos, conocidos como heurísticos, capaces de encontrar soluciones de buena calidad y que en muchos casos son muy cercanas a la solución óptima o la mejor conocida.

La palabra “*heurística*” proviene del griego *heuriskein*, que significa “*encontrar*” o “*descubrir*”. Por lo que se puede definir que los métodos heurísticos son métodos de solución basados en procedimientos conceptualmente simples con el objetivo de buscar soluciones de buena calidad a problemas difíciles, de un modo sencillo y eficiente. Estos métodos imitan fenómenos simples observados en la naturaleza y tratan de adaptar el comportamiento de diferentes especies a soluciones de problemas altamente complejos mediante optimización.

Por lo tanto, una alternativa para sobrellevar la problemática anteriormente planteada es mediante el desarrollo de un algoritmo heurístico para optimizar los factores de relajación durante el proceso iterativo de la solución numérica, con la finalidad de asegurar la convergencia del problema (caso de prueba) en el menor tiempo posible de computó.

1.3 Antecedentes (estado del arte)

La convergencia requerida durante la modelación de los fenómenos de transferencia de calor y dinámica de fluidos es difícil de obtener, especialmente en los casos no-lineales como ya se había mencionado. Aplicar el concepto de bajo-relajación ha ayudado para lograr la convergencia de la solución numérica para algunos casos específicos, sin embargo en la mayoría de las ocasiones los parámetros de relajación requeridos para las múltiples variables son producto de la experiencia individual de los desarrolladores de programas de cómputo.

La literatura sobre la convergencia y optimización de recursos computacionales para la solución de algoritmos en la dinámica de fluidos computacional (CFD, por sus siglas en inglés) es escasa, así como también la que existe es enfocada a problemas específicos.

Uno de los pioneros en el desarrollo de métodos de aceleración de la convergencia en simulaciones de CFD (Computational Fluid Dynamics) fue Brandt

(1977), quien planteó la técnica de aceleración de multimallas. Esta técnica de multimalla es una de las alternativas que se tienen para mejorar el tiempo computacional en la solución de grandes sistemas de ecuaciones algebraicas, la cual, más que una técnica es una herramienta para acelerar la convergencia de los métodos iterativos que se basa en la modificación de la malla numérica para poder dar solución al problema de divergencia.

Saad y Schultz (1986), emplearon un método iterativo llamado Generalized Minimal Residual algorithm (GMRES), para minimizar en cada paso el vector normalizado de residuales, en la solución de un sistema de ecuaciones lineales no simétrico; logrando reducir hasta un 25% el número de multiplicaciones requeridas para dar solución al sistema de ecuaciones, para llegar a la convergencia.

Ryoo et. al., (1999) plantearon algoritmos donde los factores de relajación se ajustan automáticamente durante la ejecución del código basándose en el comportamiento de los residuales, mediante la técnica de lógica difusa.

Dragojlovic et al., (2004) propusieron factores de relajación óptimos en la solución numérica de problemas de CFD, la aceleración de la convergencia la consiguieron mediante la técnica de lógica difusa, orientando la bajo-relajación en las ecuaciones discretizadas de Navier-Stokes durante la ejecución del código. Este método fue evaluado en la solución de problemas de convección natural en cavidades cuadradas. Se comprobó la eficiencia computacional utilizando este método propuesto en comparación con la utilización de factores de relajación constantes, se obtuvo un 35% de reducción en el tiempo para el fenómeno de convección natural.

Sudhakar et al., (2010) proponen una estrategia mediante un enfoque heurístico basado en la Constructal theory, la optimización se basa en un parámetro geométrico para encontrar la óptima configuración geométrica en la solución numérica, utilizando el problema de convección natural en régimen de flujo laminar conjugada en una cavidad rectangular, los autores encuentran que existe una desviación máxima del 14% entre los resultados experimentales y los numérico.

Bonilla et al., (2011) proponen un algoritmo genético para optimizar el tiempo computacional en la modelación termo-hidráulica del flujo agua-vapor del sistema DISS (Direct Solar Steam). Los autores emplean la heurística enfocándose en consideraciones temporales y espaciales así como también implementan un detector de oscilaciones durante la simulación, con lo cual logran detectar en qué momento se presenta la no-linealidad para evitar la divergencia. Con la técnica implementada lograron reducir hasta en un 85% el tiempo computacional y encontrar la convergencia al problema planteado.

En 2018 J. Xamán et al., evalúan la velocidad del tiempo de convergencia (tiempo de CPU) para la solución numérica de la ecuación de transferencia radiativa (ETR), con esquemas de alto-orden (HO) y alta-resolución (HR), aplicando el método weighting-factor normalizado (NWF), el método es utilizado para incorporar los esquemas de solución en la discretización del modelo matemático, comparan la velocidad el tiempo de convergencia del NWF con respecto a la velocidad de la técnica deferred-correction (DC). Obtienen mejoras del tiempo de convergencia utilizando el método NWF del 23.1, 12.6 y 56.1 % pero depende del esquema utilizado.

En 2018 J. Xamán et al., también proponen la incorporación de un factor de relajación modificado para acelerar la solución numérica de la ecuación de transferencia radiativa (RTE) usando el método NWF para implementar varios esquemas de resolución de alto orden, este método es conocido como el método del X-factor, los resultados obtenidos utilizando este método les dan reducción del tiempo de CPU entre 20 y 211% para la emisividad que es una de las variables de estudio.

1.4 Objetivo general

Determinar los factores de relajación (F_{relax}) cercanos a los óptimos mediante la aplicación de una heurística computacional, para la modelación de transferencia de calor convectiva.

1.4.1 Objetivos específicos

- Implementar un algoritmo en base a una heurística computacional, con información de valores residuales de las ecuaciones discretas y factores de relajación.
- Desarrollar un código computacional para la solución numérica de transferencia de calor convectiva en cavidades bidimensionales.
- Acoplar el algoritmo heurístico al código numérico con procesamiento distribuido para aplicarse al problema planteado.
- Obtener los factores de relajación cercanos a los óptimos para problemas de transferencia de calor por convección.

1.5 Contribución de la tesis

En base a la revisión de la literatura se puede observar que no existen trabajos donde se haya implementado como solución a los problemas de convergencia en problemas de CFD los métodos heurísticos, por lo que se presenta un nicho de oportunidades para poder tratar esta problemática mediante la aplicación de dichos métodos.

La aportación de esta tesis consiste en la aplicación de Recosido Simulado (RS) (que está dentro de los algoritmos heurísticos) como una alternativa para dar solución a la problemática de obtener los mejores valores para los factores de relajación que ayudan a la convergencia del problema planteado (caso de prueba).

1.6 Organización de la tesis

Este trabajo de tesis se encuentra organizado en cuatro capítulos, a continuación se describen brevemente el contenido de cada uno de ellos:

En el capítulo I, se describe la importancia del estudio desarrollado, se mencionan algunos de los problemas donde se puede presentar mayor dificultad en la convergencia para obtener la solución, así como también, se indica el caso de prueba a resolver y la alternativa de solución con la cual se aborda dicha problemática, se presenta un estado del arte donde se muestran los antecedentes que existen relacionados al estudio planteado, por último se describen los objetivos.

En el capítulo II, se describe el problema hidrodinámico (caso de prueba), se muestra el modelo físico y matemático que rige a dicho problema, la metodología de solución del mismo problema hidrodinámico,

En el capítulo III, se describe el algoritmo de Recocido Simulado, así como también, se muestra la adaptación de dicho algoritmo al problema planteado, es decir, se describe la metodología para dar solución al objetivo general basándose en el algoritmo antes descrito.

En el capítulo IV, se presenta la descripción del equipo utilizado, además, de los resultados experimentales del algoritmo de recocido simulado y el análisis de los mismos.

En el capítulo V, en éste capítulo se muestran las conclusiones obtenidas a partir del análisis de resultados y se plantean las recomendaciones para trabajos futuros en base al trabajo realizado.

CAPITULO II MARCO TEÓRICO

2.1 Descripción del problema (casos de prueba)

2.1.1 Introducción

El flujo de fluidos y transferencia de calor son fenómenos presentes en la naturaleza y en aplicaciones de ingeniería. La ecuación que representa este tipo de fenómenos es la ecuación general de convección-difusión la cual presenta numerosas aplicaciones en ingeniería y ciencias aplicadas. Por ejemplo, los modelos de convección natural que se modelan a través de dicha ecuación. Estos modelos de convección presentan muchas aplicaciones en situaciones reales, por ejemplo, cuando se trata de problemas de climatización en el área de arquitectura, de diseño de ventanas de doble vidrio etc. Debido a esto, la importancia del estudio de este tipo de fenómenos.

2.1.2 Caso 1: Convección-conducción

Variación de temperatura en estado transitorio $T(x, y, t)$, en un dominio en dos dimensiones, el cual tiene propiedades termo físicas constantes: $\rho = 2702 \text{ kg/m}^3$, $C_p = 903 \text{ J/kg}^\circ\text{C}$ y $\lambda = 237 \text{ W/m}^\circ\text{C}$. Se considera que la variable se transporta por convección y conducción de calor, y que a través del medio no existe generación de calor y que las componentes de velocidad (u, v) son constantes en cualquier punto del dominio con $u = 2 \text{ m/s}$ y $v = 2 \text{ m/s}$. El medio es geoméricamente cuadrado con una longitud de un metro $H = 1 \text{ m}$, el cual está sujeto a condiciones de frontera como se muestra en Figura 2.1.

Modelo Físico problema convección-conducción

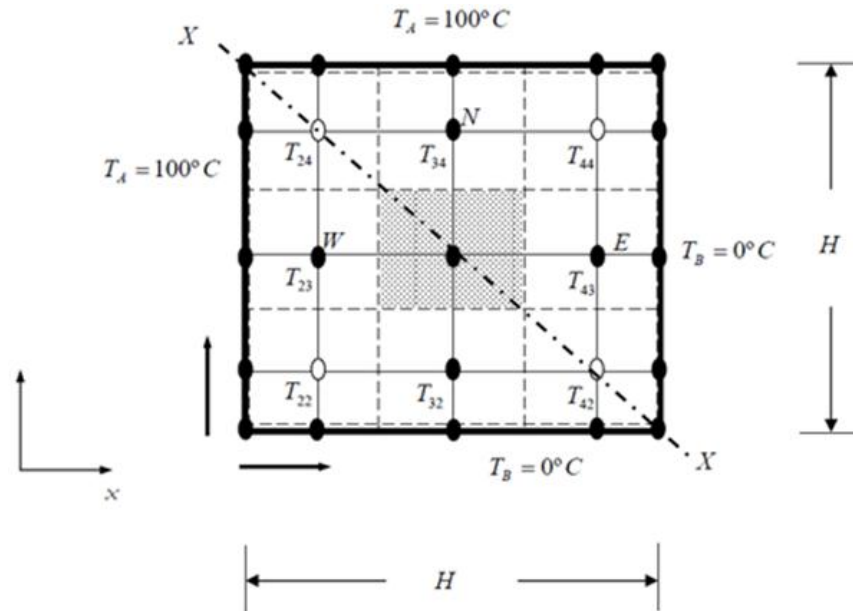


Figura 2.1. Modelo físico para el problema de convección-conducción de calor 2D, (J. Xamán et al. 2016)

2.1.3 Modelo matemático para el problema de convección-conducción

El modelo matemático para el caso 1 parte de la *ecuación general de convección difusión para la variable de interés ϕ en el sistema cartesiano* (ecuación 2.1), que en este caso la variable de interés es la temperatura (T), posteriormente mediante la técnica de volumen finito se realiza la discretización del modelo matemático planteado, la cual se representa en (ecuación 2.2) como la ecuación discreta en notación de coeficientes agrupados.

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(\rho u\phi)}{\partial x} + \frac{\partial(\rho w\phi)}{\partial y} = \frac{\partial}{\partial x} \left(\Gamma \frac{\partial\phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\Gamma \frac{\partial\phi}{\partial y} \right) + S \quad (2.1)$$

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (2.2)$$

2.1.4 Caso 2: Problema hidrodinámico

El problema a tratar consiste en resolver las ecuaciones de continuidad y momentum en 2D en una cavidad cuadrada, sujeta a condiciones de frontera de primera clase.

El problema se define en régimen de flujo laminar e incompresible en una cavidad llena de un fluido con propiedades físicas de: viscosidad dinámica, $\mu = 1.817 \times 10^{-5} \text{ kg/(m.s)}$ y densidad $\rho = 1.2047 \text{ kg/m}^3$. La cavidad es cuadrada de ancho $Hx = Hy = 1 \text{ m}$ y su pared superior se desplaza con una velocidad uniforme $U_0 = 1.508 \times 10^{-3} \text{ m/s}$ correspondiente a un número de Reynolds ($Re = \rho Hx U_0 / \mu$) de 100. El problema es conocido como “*Driven-Cavity Problem*”. El problema se usa como problema de referencia para probar y evaluar técnicas numéricas (como es el caso en el presente trabajo). Ghia et al. En 1982 publicaron los resultados de referencia. En la Figura 2.2 se muestra el modelo físico del problema con sus respectivas condiciones de frontera.

Modelo Físico problema hidrodinámico

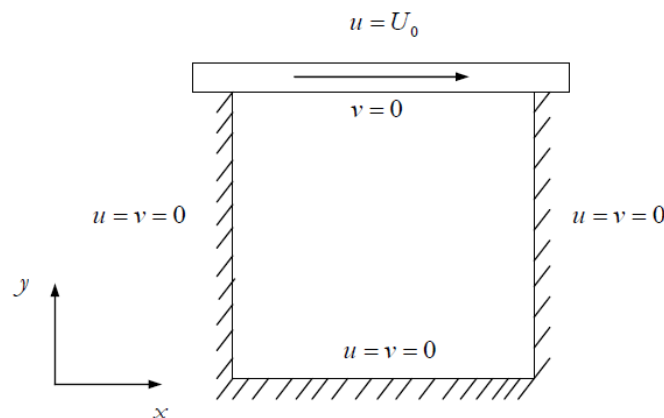


Figura 2.2. Modelo físico para el problema hidrodinámico, (J. Xamán et al. 2016)

Tabla 2.1 Propiedades físicas del problema hidrodinámico

ρ (kg/m ³)	μ (kg/(m.s))	Número de Reynolds
1.2047	1.817×10^{-5}	400

Tabla 2.2 Condiciones iniciales del problema hidrodinámico

$U_0 = (m/s)$
1.508×10^{-3}

2.1.5 Modelo matemático del problema hidrodinámico

En los problemas de dinámica de fluidos se ven involucradas las componentes de velocidad y el gradiente de presión, las ecuaciones que gobiernan este fenómeno hidrodinámico son las ecuaciones de conservación de masa y momentum (Ecs. (2.3)-(2.5) con $F_x = F_y = 0$), las cuales serán resueltas con el uso del algoritmo SIMPLE y el esquema Híbrido o Ley de Potencia. Se considera el caso bidimensional, para un fluido Newtoniano e incompresible en régimen laminar. Por lo tanto, las ecuaciones resultantes adaptadas de la ecuación general de convección-difusión son:

Ecuación de masa:

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \quad (2.3)$$

Ecuación Momento en x:

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho uu)}{\partial x} + \frac{\partial(\rho vu)}{\partial y} = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) - \frac{\partial P}{\partial x} \quad (2.4)$$

Ecuación de Momento en y:

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho vv)}{\partial y} = \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) - \frac{\partial P}{\partial y} \quad (2.5)$$

A partir de las ecuaciones anteriores se realizan las consideraciones correspondientes de acuerdo al planteamiento del problema.

Consideraciones:

- 1) Bidimensional $f(x, y, z) = f(x, y)$
- 2) Fluido $f(u, v, w) = f(u, v)$
- 3) Estado transitorio $f(t)$

2.2 Metodología de solución

Para dar solución al problema planteado por el Método de Volumen Finito (MVF) se realizan tres pasos principales:

- 1) Generación de la malla, que consiste en dividir el dominio del problema en Volúmenes de Control.
- 2) Discretización, convierte las ecuaciones integrales en un sistema de ecuaciones algebraicas.
- 3) Solución del sistema de ecuaciones algebraicas por un método eficiente.

2.2.1 Generación de la malla computacional

Considerando que la solución numérica de ecuaciones diferenciales parciales (EDPs) requiere de la discretización de las ecuaciones sobre un dominio discreto (malla computacional), el proceso de discretización reemplaza las ecuaciones diferenciales parciales con un sistema de ecuaciones algebraicas discretas (como se verá más adelante). Así, al utilizar una técnica numérica para la solución de este sistema, es necesario discretizar el campo usando una malla, la cual es una colección de puntos discretos distribuida sobre un campo de cálculo, es decir, se divide el modelo físico en volúmenes de control

de manera ordenada y cubriendo todo el dominio como se muestra en la Figura 2.3, de tal manera que un número finito de puntos o de nodos se ubican entre cada una de las fronteras (Este, Oeste, Norte, Sur, según sea el caso), uno para cada volumen de control.

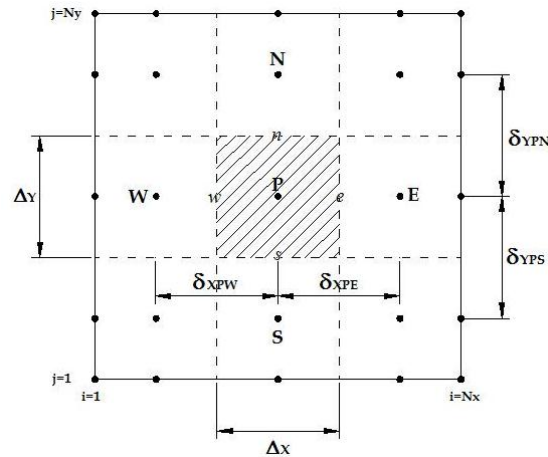


Figura 2.3 Representación de volúmenes de control en 2-D.

2.2.2 Concepto de malla desplazada

Uno de los pasos para el acople de las ecuaciones de continuidad y momentum es el uso de las mallas desplazadas o superpuestas las cuales son utilizadas al acoplar las ecuaciones de conservación de masa y momento, se superponen de dos a tres mallas para dos y tres dimensiones respectivamente, la malla descrita anteriormente es conocida como malla principal en donde se representan las variables escalares como presión, temperatura, energía cinética etc. según sea el caso. Las otras mallas se desplazan de la malla principal en direcciones x, y de tal forma que las fronteras de sus volúmenes de control coinciden con los puntos nodales de la malla principal, estas almacenas los componentes de velocidad u, v , respectivamente, como se muestra en la Figura siguiente.

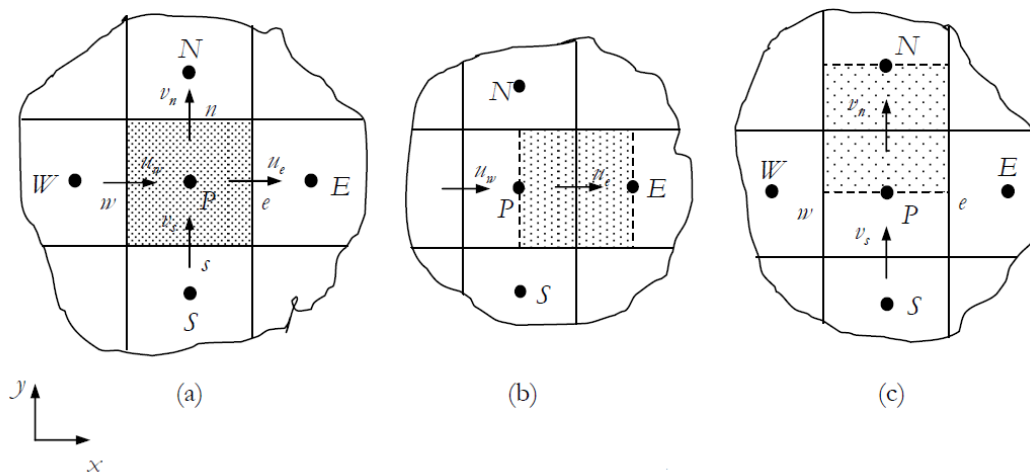


Figura 2.4 Representación de mallas superpuestas: a) VC para variables escalares, b) VC para u_e y c) VC para v_n , (J. Xamán et al. 2016)

2.2.3 Discretización del modelo matemático

El problema planteado es un problema convectivo-difusivo en 2D, por lo que se requiere tener un buen esquema de interpolación para tomar en cuenta los términos convectivos, esta sección considera cuatro esquemas de bajo orden para poder tratar los términos convectivos, estos se plantean en 1D para después considerarlos de manera general en la ecuación discretizadas por el MFV.

2.2.3.1 Esquemas de interpolación de bajo orden

Propiedades de los esquemas de interpolación

Para asegurar que un esquema de interpolación represente el fenómeno físico de manera adecuada éste debe cumplir los siguientes puntos:

1. *Conservativo*: Para asegurar la conservación de la variable en todo el dominio, el flujo de la variable que sale a través de un volumen de

control debe ser igual al flujo que entra en el volumen de control adyacente.

2. *Acotado*: Una característica deseable para satisfacer la “Acotabilidad” es tener un sistema diagonalmente dominante y una condición para ello es el criterio de Scarborough (1958):

$$\frac{\sum |a_{\text{vecinos}}|}{|a_p|} = \begin{cases} \leq 1 & \text{En todos los nodos} \\ < 1 & \text{Almenos en un nodo} \end{cases}$$

3. *Transportividad*: Que relacione la influencia de la intensidad del flujo o la intensidad de la convección a la difusión (P_e). El número de Peclet se define como:

$$\text{Número de Peclet } (P_e) = \frac{F}{D} = \frac{\rho u}{\Gamma / \delta x} \quad (2.6)$$

2.2.3.2 Esquema centrado

Para realizar este análisis se considera el volumen de control que se muestra en la Figura 2.5, el modelo matemático a discretizar es la ecuación de convección difusión en una dimensión.

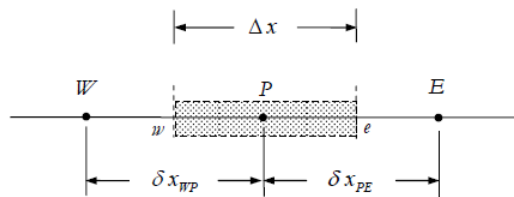


Figura 2.5 Volumen de control sujeto a análisis.

Ecuación de convección-difusión en 1D:

$$\frac{\partial(\rho u \phi)}{\partial x} = \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) \quad (2.7)$$

Integrando la ecuación (2.7) espacialmente sobre los límites geométricos del volumen de control:

$$\int_w^e \frac{\partial(\rho u \cdot \phi)}{\partial x} dx = \int_w^e \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) dx \quad (2.8)$$

resolviendo las integrales de la ecuación anterior:

$$(\rho u \cdot \phi)|_w^e = \left(\Gamma \frac{\partial \phi}{\partial x} \right) \Big|_w^e \quad (2.9)$$

evaluando los límites:

$$(\rho u \cdot \phi)_e - (\rho u \cdot \phi)_w = \left(\Gamma \frac{\partial \phi}{\partial x} \right) \Big|_e - \left(\Gamma \frac{\partial \phi}{\partial x} \right) \Big|_w \quad (2.10)$$

Definiendo F como intensidad convectiva y D como intensidad difusiva:

$$F_x = \rho u \quad (2.11)$$

$$D_x = \frac{\Gamma}{\delta_x} \quad (2.12)$$

Sustituyendo las ecuaciones anteriores (2.11 y 2.12) en 2.10, y empleando una diferencia centrada para los términos difusivos, se tiene que la ecuación algebraica que representa la variable ϕ , está dada por:

$$F_e \phi_e - F_w \phi_w = D_e (\phi_e - \phi_P) - D_w (\phi_P - \phi_w) \quad (2.13)$$

Definiendo una interpolación de esquema centrado, la cual se usa para calcular los valores de la variable ϕ en las fronteras del volumen de control como una malla aritmética o una interpolación lineal, se obtienen los siguientes términos:

$$\phi_e = \frac{1}{2} (\phi_P + \phi_E) \quad (2.14)$$

$$\phi_w = \frac{1}{2} (\phi_W + \phi_P) \quad (2.15)$$

Sustituyendo la interpolación del esquema centrado en 2.13 para los términos convectivos, se llega a,

$$\frac{F_e}{2}(\phi_P + \phi_E) - \frac{F_w}{2}(\phi_W + \phi_P) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W) \quad (2.16)$$

Acomodando términos,

$$\left[\frac{F_e}{2} + D_e - \frac{F_w}{2} + D_w\right] \phi_P = \left[D_e - \frac{F_e}{2}\right] \phi_E + \left[D_w + \frac{F_w}{2}\right] \phi_W \quad (2.17)$$

El siguiente paso es hacer que la ecuación (2.17) sea conservativa es decir que satisfaga la ecuación de conservación de masa, esto es introduciendo la ecuación de conservación discretizada, cual se representa con la siguiente expresión:

$$(F_e - F_w)\phi_P = 0 \quad (2.18)$$

Introduciendo la ecuación (2.18) en (2.17) y factorizando y acomodando términos se llega a la siguiente expresión:

$$\left[\left(D_e - \frac{F_e}{2}\right) + \left(D_w + \frac{F_w}{2}\right) + (F_e - F_w)\right] \phi_P = \left(D_e - \frac{F_e}{2}\right) \phi_E + \left(D_w + \frac{F_w}{2}\right) \phi_W \quad (2.19)$$

Finalmente la ecuación algebraica de convección-difusión en una dimensión puede compactarse en notación de coeficientes agrupado como:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (2.20)$$

donde

$$a_E = D_e - \frac{F_e}{2}, \quad a_W = D_w + \frac{F_w}{2} \quad (2.21)$$

$$a_P = a_E + a_W + (F_e - F_w) \quad (2.22)$$

El esquema centrado no toma la dirección del flujo de la variable en la interface, se recomienda usar este esquema para problemas de difusión pura, ya que aplicado a otros problemas es muy probable obtener resultados

físicamente no reales o incluso la matriz puede no ser diagonalmente dominante y convertirse en una matriz singular (sistemas de ecuaciones algebraicas sin solución).

2.2.3.3 Esquema Upwind (corrientes arriba)

Un remedio para sobrellevar las dificultades encontradas con el esquema centrado es usar un esquema (upwind). Este esquema propone una mejor aproximación para la variable en la interface al tomar en cuenta la dirección de la corriente principal del flujo.

$$\phi_E = \begin{cases} \phi_P & \text{si } F_e > 0 \\ \phi_E & \text{si } F_e < 0 \end{cases} \quad (2.23)$$

$$\phi_w = \begin{cases} \phi_W & \text{si } F_w > 0 \\ \phi_P & \text{si } F_w < 0 \end{cases} \quad (2.24)$$

La ecuación (2.10) puede escribirse como:

$$F_e \phi_P - F_w \phi_W = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_w) \quad (2.25)$$

La cual puede reordenarse de la siguiente manera,

$$(D_e + D_w + F_e)\phi_P = (D_w + F_w)\phi_W + D_e\phi_E \quad (2.26)$$

Al introducir la ecuación de conservación de masa discretizada, para cuando la dirección de flujo es positiva, se obtiene:

$$[(D_w + F_w) + D_e + (F_e - F_w)]\phi_P = (D_w + F_w)\phi_W + D_e\phi_E \quad (2.27)$$

De manera análoga para cuando el flujo es una dirección negativa, se llega a:

$$[(D_e - F_e) + D_w + (F_e - F_w)]\phi_P = (D_e - F_e)\phi_e + D_w\phi_W \quad (2.28)$$

De manera compacta se llega a la ecuación en notación de coeficientes agrupados, las ecuaciones (2.27 y 2.28) se escriben como:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (2.29)$$

$$a_P = a_E + a_W + (F_e - F_w) \quad (2.30)$$

$$a_E = D_e, a_W = D_w + F_w \quad \text{para} \quad F_w > 0, \quad F_e > 0 \quad (2.31)$$

$$a_E = D_e - F_e, a_W = D_w \quad \text{para} \quad F_w < 0, \quad F_e < 0 \quad (2.32)$$

Los coeficientes a_E y a_W se pueden expresar de manera general para cualquier dirección de flujo por medio de la función máxima, esto es:

$$a_E = D_e + \max(-F_e, 0) \quad (2.33)$$

$$a_W = D_w + \max(F_w, 0) \quad (2.34)$$

Para este esquema, se puede demostrar que los coeficientes son todos siempre positivos, por lo tanto se considera como un esquema estable.

2.2.3.4 Esquema Híbrido

Este esquema toma las características del esquema centrado y el esquema upwind, combinándolas en intervalos convenientes.

En la Figura 2.6 se observan las siguientes propiedades de la variación exacta:

$$\text{Para } P_{e_e} \rightarrow \infty: \frac{a_e}{D_e} \rightarrow 0$$

$$\text{Para } P_{e_e} \rightarrow -\infty: \frac{a_e}{D_e} \rightarrow -P_{e_e}$$

$$\text{Para } P_{e_e} = 0: \frac{a_e}{D_e} = \frac{(1-P_{e_e})}{2}$$

Una aproximación al comportamiento de la solución exacta, es tomar en cuenta los intervalos donde el coeficiente a_E se comporta bien con el esquema centrado (valores del número de Peclet bajos) y cuando el coeficiente a_E tiende

al comportamiento del esquema upwind. La siguiente Figura muestra el comportamiento de la solución exacta y las líneas tangentes que forman el esquema híbrido.

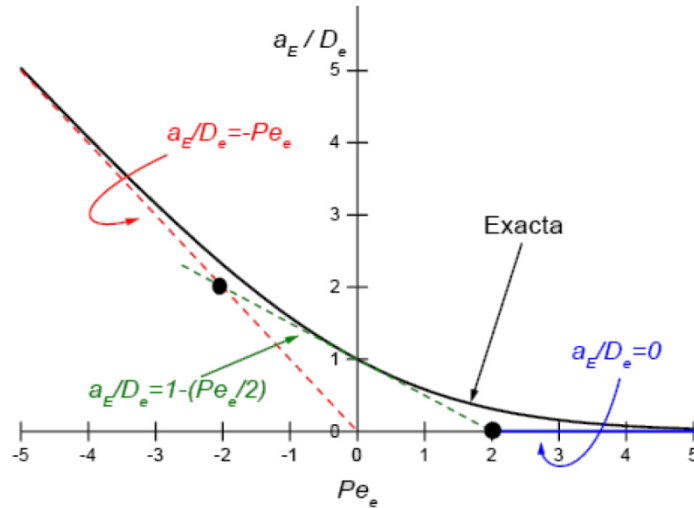


Figura 2.6 Variación de a_E con el número de Peclet (P_e), (J. Xamán et al. 2016)
Considerando la Figura anterior se llega a las siguientes expresiones:

$$\frac{a_E}{D_e} = -P_{e_e} \quad \text{para} \quad P_{e_e} < -2 \quad (2.35)$$

$$\frac{a_E}{D_e} = 1 - \frac{P_{e_e}}{2} \quad \text{para} \quad -2 \leq P_{e_e} \leq 2 \quad (2.36)$$

$$\frac{a_E}{D_e} = 0 \quad \text{para} \quad P_{e_e} > 2 \quad (2.37)$$

Las ecuaciones anteriores pueden compactarse en una sola expresión, esto es,

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (2.38)$$

$$a_E = \max(-F_e, (D_e + \frac{F_e}{2}), 0) \quad (2.39)$$

$$a_W = \max(F_w, (D_w + \frac{F_w}{2}), 0) \quad (2.40)$$

$$a_P = a_E + a_W + (F_e - F_w) \quad (2.41)$$

El esquema híbrido es idéntico al esquema centrado para bajos números de Peclet $-2 \leq P_{e_e} \leq 2$. Fuera de ese intervalo el esquema híbrido se reduce al esquema upwind en el cual el término de difusión es agrupado a cero.

2.2.3.5 Esquema Ley de Potencia

Se observa en la Figura 2.7 que la desviación del esquema híbrido con respecto a la curva exacta es grande en $Pe \pm 2$, esta desviación provoca que se agrupen los efectos de difusión prematuramente a cero tan pronto $|Pe|$ excede 2.

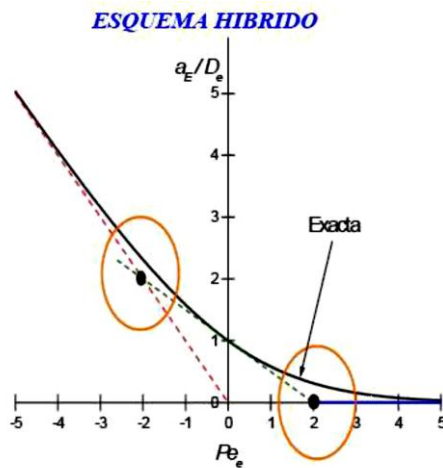


Figura 2.7. Desviación del esquema híbrido respecto al valor exacto de a_E/D_e , (J. Xamán et al. 2016).

Las relaciones propuestas por Patankar para el coeficiente a_E/D_e , son definidas en los siguientes intervalos:

$$\frac{a_E}{D_E} = \begin{cases} -P_{e_e} & \text{para } P_{e_e} < -10 \\ (1 + 0.1 - P_{e_e})^5 - P_{e_e} & \text{para } -10 \leq P_{e_e} < 0 \\ (1 + 0.1P_{e_e})^5 & \text{para } 0 \leq P_{e_e} < 10 \\ 0 & \text{para } P_{e_e} > 10 \end{cases} \quad (2.42)$$

Los resultados de la ecuación (2.42) se muestran en la Figura 2.8, en la cual se aprecia que las desviaciones observadas con el esquema híbrido ya no existen.

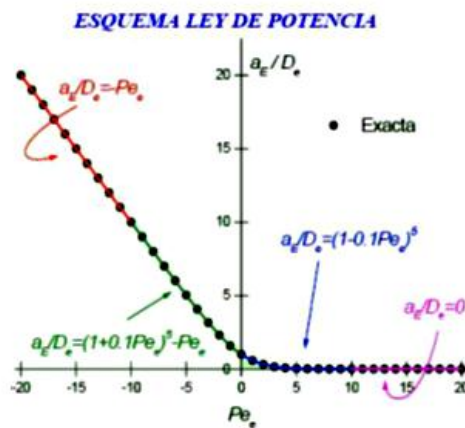


Figura 2.8 Función a_E/D_e para el esquema ley de potencia, (J. Xamán et al. 2016).

La forma final de los coeficientes a_E y a_W del esquema de ley de potencia es:

$$a_E = D_e \max[0, (1 - 0.1|F_e/D_e|)^5] + \max[-F_e, 0] \quad (2.43)$$

$$a_W = D_w \max[0, (1 - 0.1|F_w/D_w|)^5] + \max[F_w, 0] \quad (2.44)$$

En 1980, Patankar compactó de forma general los esquemas numéricos presentados anteriormente, al cual le llamo la formulación generalizada para la ecuación de convección-difusión. Para poder obtener la formulación generalizada, Patankar definió una función en términos del número de Peclet para cada uno de los esquemas anteriores. Así, finalmente la ecuación algebraica en notación de coeficientes agrupados para el problema

unidimensional, en estado permanente y sin término fuente se puede escribir como,

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (2.45)$$

$$a_E = D_e A(|P_{e_e}|) + \max[-F_e, 0] \quad (2.46)$$

$$a_W = D_w A(|P_{e_e}|) + \max[F_w, 0] \quad (2.47)$$

$$a_P = a_E + a_W + (F_e - F_w) \quad (2.48)$$

Los flujos convectivos (F), términos difusivos (D) y número de Peclet (P_e) a través de las caras del volumen de control son:

$$F_e = (\rho u)_e \quad , \quad F_w = (\rho u)_w \quad (2.49)$$

$$D_e = \frac{\Gamma_e}{\delta x_{PE}} \quad , \quad D_w = \frac{\Gamma_w}{\delta x_{PW}} \quad (2.50)$$

$$P_e = F_e/D_e \quad , \quad P_w = F_w/D_w \quad (2.51)$$

La función $A(|P_{e_e}|)$ es una función que depende del esquema numérico utilizado. En la Tabla 2.3 se muestran las equivalencias de la función $A(|P_{e_e}|)$ para los diferentes esquemas discutidos previamente (Patankar, 1980)

Tabla 2.3 Función $A(|P_{e_e}|)$, con respecto al esquema numérico

Esquema numérico	$A(Pe)$
Centrado	$1 - 0.5 Pe $
Upwind	1
Exacto o Exponencial	$\frac{ Pe }{e^{ Pe } - 1}$
Híbrido	$\max[0, (1 - 0.5 Pe)]$
Ley de potencia	$\max[0, (1 - 0.1 Pe)^5]$

En resumen, de estos esquemas convencionales se puede decir (J. Xamán et al. 2016):

Esquema Centrado: usa el promedio de los dos valores nodales más cercanos a la frontera del volumen de control para aproximar a la variable en la interface. Funciona bien para problemas a bajas velocidades pero no es aconsejable para situaciones altamente convectivas, ya que no representa adecuadamente el transporte convectivo de las propiedades.

Esquema Upwind: aproxima el valor de la variable en la frontera del VC con el valor nodal inmediato a la frontera, según el sentido de la velocidad. Presenta resultados físicamente aceptables pero con baja exactitud. Para mejorar exactitud de los resultados se tiene que usar una malla más densa. Tiene un buen comportamiento para la convergencia, ya que no es oscilatorio.

Esquema Híbrido: combina las características del esquema centrado y del esquema de upwind. Usa el esquema centrado para velocidades bajas y para velocidades elevadas utiliza las características del esquema upwind.

Esquema de Ley de Potencia: es una modificación del esquema híbrido con base a la solución exacta unidimensional para mejorar la exactitud.

2.2.4 Algoritmo SIMPLE

Dos cosas importantes a considerar en las ecuaciones anteriores son: (1) la ecuación de momento tiene un término convectivo, el cual es altamente no-lineal y (2) las ecuaciones de momentum y continuidad están fuertemente acopladas debido a las componentes de velocidad que aparecen en ellas, lo cual forma un sistema de ecuaciones diferenciales parciales. Por lo que, una complejidad importante se presenta al estimar el rol de la presión en las ecuaciones de momento, ya que no existe una ecuación de transporte para la presión. También, se observa que en las ecuaciones de momentum, el término de gradiente de presión y otras fuerzas conservativas (F_i) forman parte del término fuente (efectos electromagnéticos, coriolis, gravitatorios, etc.) de la ecuación general de convección-difusión, (J. Xamán et al. 2016).

Este algoritmo es una técnica de solución secuencial utilizada para acoplar las ecuaciones de conservación de masa y movimiento, donde se usan las variables de velocidad y presión, por lo que conlleva a ciertos problemas a la hora de trabajar con el gradiente de presión en las ecuaciones de movimiento, por ello Patankar (1980) mostró que al discretizar dichas ecuaciones en los mismos nodos computacionales provocaba una distribución de presión oscilatoria no correspondiente a la solución real, por lo que se conduce a la utilización de mallas desplazadas, como se vio en la sección anterior.

2.2.4.1 Gradiente de Presión

Cuando se trabaja con fenómenos en donde interviene el flujo de fluidos, puede que el campo de presión sea desconocido, cuando el flujo es incompresible, la densidad se considera constante y la ecuación de continuidad puede prescindir de ella, para ello es factible suponer un campo de presión inicial para cálculos con métodos iterativos y estimar el campo de velocidades.

Tomando en cuenta que la representación del gradiente de presión de la ecuación de momento en dirección- x en el nodo “P” ($-\frac{\partial P}{\partial x}$) integrado sobre el volumen de control da como resultado la caída de presión ($P_w - P_e$), lo cual representa la presión ejercida sobre la frontera oeste menos la presión ejercida sobre la frontera este del volumen de control, es decir, la fuerza por unidad de área ejercida por las presiones sobre el VC.

Debido a que las presiones son calculadas en la malla principal no se cuenta con información sobre ella en las fronteras de los volúmenes de control, por lo que se realiza una interpolación lineal con los valores de los nodos vecinos, de tal manera que si la malla usada es regular el gradiente de presión puede tomarse como:

$$P_w - P_e = \frac{P_W + P_P}{2} - \frac{P_P + P_E}{2} = \frac{P_W - P_E}{2} \quad (2.52)$$

De la ecuación anterior se observa que el gradiente de presión es la diferencia de presión entre dos puntos alternantes lo cual no resulta lógico, ya que en ciertos casos esta diferencia daría cero e indicaría que la presión es constante o uniforme y esto no es cierto, debido a esta complicación es que se recurre al uso de mallas desplazadas con lo que ahora se tiene información de la presión en las fronteras del volumen de control, evitando aproximaciones y haciendo los balances de presión inmediatos.

2.2.4.2 Formulación del algoritmo SIMPLE

Para acoplar la determinación de la variable presión con la resolución de las ecuaciones de hidrodinámica Patankar y Spalding (1972) propusieron el algoritmo SIMPLE por el acrónimo en inglés Semi-Implicit Method for Pressure-Linked Equations (Método semi implícito para el acople de las ecuaciones de presión).

La filosofía del método es resolver las ecuaciones de conservación, utilizando el concepto de malla desplazada. El hecho de desplazar las mallas para las velocidades implica que la interface de sus VC se relacione con los puntos nodales de la malla principal y sobre éstos se tiene la información almacenada de la presión, por lo tanto, los balances de presión son inmediatos. En la Figura 2.9, se muestran las mallas superpuestas en el plano bidimensional: malla principal para las variables escalares (●), malla desplazada en dirección - x para la componente de velocidad horizontal (□) y la malla desplazada en dirección - y para la componente de velocidad vertical (Δ). Se puede observar que las coordenadas de la malla principal se pueden ubicar a través del punto (I, J), mientras que la malla desplazada para la velocidad-u se ubica por la coordenada (i, J) y la malla desplazada para la velocidad-v se localiza por la coordenada (I, j). A partir de aquí, esta nomenclatura será usada cuando sea necesario para ubicar las variables escalares o vectoriales. También, la malla desplazada en la dirección horizontal o vertical, se le denotará como malla-u o malla- v, respectivamente, (J. Xamán et al., 2016).

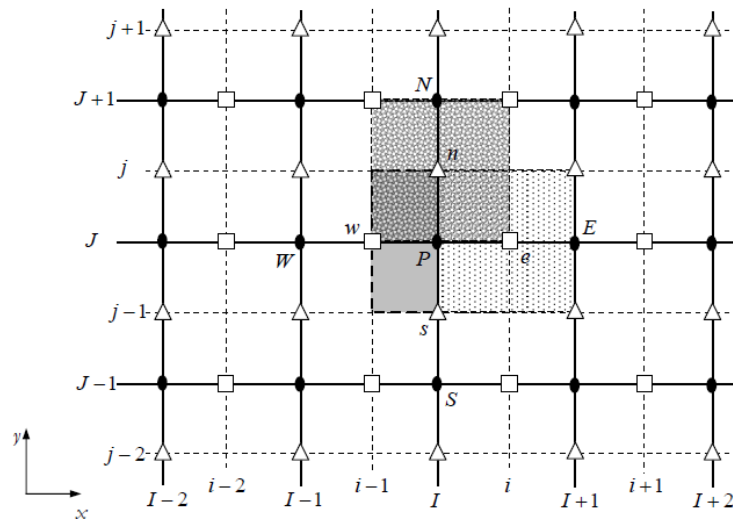


Figura 2.9 Malla principal (●), malla desplazada en dirección - x (□) y malla desplazada en dirección - y (Δ), (J. Xamán et al., 2016).

La estructura del algoritmo SIMPLE está compuesta de dos partes: (1) la suposición de un campo de presiones lo cual facilita la obtención de una distribución de velocidades, (2) corrección de estas velocidades de manera iterativa hasta llegar a una solución correcta; a continuación se describe el algoritmo SIMPLE paso a paso:

Como primer paso se integra el término de presión sobre el volumen e control (momento-x):

$$\int_s^n \int_P^E \left(\frac{\partial P}{\partial x}\right) dx dy = \int_s^n dy \int_P^E \frac{\partial P}{\partial x} dx = -\Delta y (P_E - P_P) \quad (2.53)$$

Análogamente para momento en “y”

$$\int_P^E \int_P^N \left(\frac{\partial P}{\partial y}\right) dx dy = \int_P^E dx \int_P^N \frac{\partial P}{\partial y} dy = -\Delta x (P_N - P_P) \quad (2.54)$$

Descomponer el término fuente de las ecuaciones de momento, de tal forma que la presión aparezca explícitamente.

$$b = -A_e (P_E - P_P) \quad (2.55)$$

$$b = -A_n (P_N - P_P) \quad (2.56)$$

donde $A_e = \Delta y$ y $A_n = \Delta x$ es el área del VC que atraviesa el flux correspondiente (convectivo y/o difusivo) para dos dimensiones.

Discretización de las ecuaciones de momento sobre sus respectivas mallas desplazadas.

Ecuación discretizada de momento-x:

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u)}{\partial t} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u u)}{\partial x} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u v)}{\partial y} dx dy dt = \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) dx dy dt \quad (2.57)$$

Resolviendo las integrales para el término temporal:

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u)}{\partial t} dx dy dt = [(\rho u)_p - (\rho u)_p^0] \Delta x \Delta y \quad (2.58)$$

Resolviendo para términos convectivos:

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u u)}{\partial x} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u v)}{\partial y} dx dy dt = [(\rho u \cdot u)_e - (\rho u \cdot u)_w] \Delta y \Delta t + [(\rho v \cdot u)_n - (\rho v \cdot v)_s] \Delta y \Delta t \quad (2.59)$$

para términos conductivos:

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) dx dy dt = \left[\mu \frac{\partial u}{\partial x} \right]_e - \left[\mu \frac{\partial u}{\partial x} \right]_w \Delta y \Delta t + \left[\mu \frac{\partial u}{\partial x} \right]_n - \left[\mu \frac{\partial u}{\partial x} \right]_s \Delta x \Delta t \quad (2.60)$$

Uniendo las ecuaciones (2.58, 2.59, 2.60 y 2.55), acomodando términos se llega a:

$$[(\rho u)_p - (\rho u)_p^0] \Delta x \Delta y + [(\rho u \cdot u)_e - (\rho u \cdot u)_w] \Delta y \Delta t + [(\rho v \cdot u)_n - (\rho v \cdot v)_s] \Delta y \Delta t = \left[\mu \frac{\partial u}{\partial x} \right]_e - \left[\mu \frac{\partial u}{\partial x} \right]_w \Delta y \Delta t + \left[\mu \frac{\partial u}{\partial x} \right]_n - \left[\mu \frac{\partial u}{\partial x} \right]_s \Delta x \Delta t - (P_E - P_P) \Delta y \quad (2.61)$$

Para simplificar la ecuación anterior, se definen los siguientes términos que ayudan a compactar dicha ecuación:

Los fluxes convectivos a través de las caras del volumen de control:

$$F_e^u = (\rho u)_E \Delta y = \frac{(\rho u)_{i,j} + (\rho u)_{i+1,j}}{2} \Delta y$$

$$F_w^u = (\rho u)_P \Delta y = \frac{(\rho u)_{i-1,j} + (\rho u)_{i,j}}{2} \Delta y \quad (2.62)$$

$$F_n^u = (\rho v)_{i,j} \Delta x = \frac{(\rho v)_{i,j} + (\rho v)_{i+1,j}}{2} \Delta x$$

$$F_s^u = (\rho v)_{i,j-1} \Delta x = \frac{(\rho v)_{i,j-1} + (\rho v)_{i+1,j-1}}{2} \Delta x$$

Los fluxes conductivos a través de las caras del volumen de control:

$$\begin{aligned}
D_e^u &= \left(\frac{\mu}{\delta x}\right)_E \Delta y = \frac{\mu_{i+1,J}^u}{(x_{i+1}^u - x_i^u)} \Delta y \\
D_w^u &= \left(\frac{\mu}{\delta x}\right)_P \Delta y = \frac{\mu_{i,J}^u}{(x_i^u - x_{i-1}^u)} \Delta y \\
D_n^u &= \left(\frac{\mu}{\delta y}\right)_{i,j} \Delta x = \frac{\mu_{i,J}^u + \mu_{i,J+1}^u + \mu_{i+1,J}^u + \mu_{i+1,J+1}^u}{4(y_{j+1}^u - y_j^u)} \Delta x \\
D_s^u &= \left(\frac{\mu}{\delta y}\right)_{i,j-1} \Delta x = \frac{\mu_{i,J-1}^u + \mu_{i,J}^u + \mu_{i+1,J-1}^u + \mu_{i+1,J}^u}{4(y_j^u - y_{j-1}^u)} \Delta x
\end{aligned} \tag{2.63}$$

Flujos totales a través de las caras de VC (Flux conductivos más flux convectivos):

$$\begin{aligned}
J_e &= \left[(\rho u u)_E - \left(\mu \frac{\partial u}{\partial x} \Big|_e \right) \right] \Delta y \\
J_w &= \left[(\rho u u)_P - \left(\mu \frac{\partial u}{\partial x} \Big|_p \right) \right] \Delta y \\
J_n &= \left[(\rho v u)_n - \left(\mu \frac{\partial u}{\partial y} \Big|_n \right) \right] \Delta x \\
J_s &= \left[(\rho v u)_s - \left(\mu \frac{\partial u}{\partial y} \Big|_s \right) \right] \Delta x
\end{aligned} \tag{2.64}$$

Sustituyendo los términos anteriores en la ecuación (2.59) se llega a la siguiente expresión:

$$\left[(\rho u)_p - (\rho u)_p^0 \right] \frac{\Delta x \Delta y}{\Delta t} + [J_e - J_w] + [J_n - J_s] + [P_E - P_P] \Delta y = \bar{S} \Delta x \Delta y \tag{2.65}$$

El cumplimiento de la ecuación de conservación de masa es prioritario en cualquier sistema físico, por lo tanto se discretiza la ecuación de conservación de masa (continuidad) dada por:

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{2.66}$$

La ecuación de continuidad (2.66) ya discretizada queda como:

$$[(\rho)_P - (\rho)_P^0] \frac{\Delta x \Delta y}{\Delta t} + (F_e^u - F_w^u) + (F_n^u - F_s^u) = 0 \quad (2.67)$$

Multiplicando la ecuación anterior por u_P

$$[(\rho)_P - (\rho)_P^0] u_P \frac{\Delta x \Delta y}{\Delta t} + F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P = 0 \quad (2.68)$$

Sumando y restando la ecuación anterior (2.68) a la ecuación (2.65) para no afectarla:

$$\begin{aligned} & [(\rho u)_P - (\rho u)_P^0] \frac{\Delta x \Delta y}{\Delta t} + [J_e - J_w] + [J_n - J_s] + [P_E - P_P] \Delta y + \left[[(\rho)_P - (\rho)_P^0] u_P \frac{\Delta x \Delta y}{\Delta t} + F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P \right] - \left[[(\rho)_P - (\rho)_P^0] u_P \frac{\Delta x \Delta y}{\Delta t} + F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P \right] = \bar{S} \Delta x \Delta y \end{aligned}$$

Simplificando a expresión anterior se llega a:

$$\begin{aligned} & [(u - u_P^0)] \rho_P^0 \frac{\Delta x \Delta y}{\Delta t} + [(J_e - F_e^u u_P) - (J_w - F_w^u u_P)] + [(J_n - F_n^u u_P) - (J_s - F_s^u u_P)] + [F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P] = \bar{S} \Delta x \Delta y \end{aligned} \quad (2.69)$$

Ahora corresponde pasar la ecuación discreta anterior a una notación de coeficiente agrupados, para ello se usa la formulación de esquema generalizado (Patankar, 1980).

$$(J_e - F_e^u u_P) = a_E (u_P - u_E) \quad (2.70a)$$

$$(J_w - F_w^u u_P) = a_W (u_W - u_P) \quad (2.70b)$$

$$(J_n - F_n^u u_P) = a_N (u_P - u_N) \quad (2.70c)$$

$$(J_s - F_s^u u_P) = a_S (u_S - u_P) \quad (2.70d)$$

al sustituir las ecuaciones del esquema generalizado en la ecuación (2.69) se obtiene:

$$[(u - u_p^0)]\rho_p^0 \frac{\Delta x \Delta y}{\Delta t} + [a_E(u_P - u_E) - a_W(u_W - u_P)] + [a_N(u_P - u_N) - a_S(u_S - u_P)] + [F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P] = \bar{S} \Delta x \Delta y \quad (2.71)$$

Finalmente simplificando y factorizando la ecuación anterior con el objetivo de llegar a la ecuación en notación de coeficientes agrupados:

$$[(u - u_p^0)]\rho_p^0 \frac{\Delta x \Delta y}{\Delta t} + a_E u_P - a_E u_E - a_W u_W + a_W u_P + a_N u_P - a_N u_N - a_S u_S + a_S u_P + [F_e^u u_P - F_w^u u_P + F_n^u u_P - F_s^u u_P] = \bar{S} \Delta x \Delta y$$

$$\left[[(u - u_p^0)]\rho_p^0 \frac{\Delta x \Delta y}{\Delta t} + a_E + a_W + a_N + a_S + (F_e^u - F_w^u + F_n^u - F_s^u) \right] u_p = a_E u_E + a_W u_W + a_N u_N + a_S u_S$$

Por lo tanto la ecuación discretizada de convección-difusión bidimensional en notación de coeficientes agrupados se representa con la siguiente expresión:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (2.72)$$

donde,

$$a_E = D_e A(|P_{e_e}|) + \max[-F_e, 0] \quad (2.73a)$$

$$a_W = D_w A(|P_{e_w}|) + \max[F_w, 0] \quad (2.73b)$$

$$a_N = D_n A(|P_{e_n}|) + \max[-F_n, 0] \quad (2.73c)$$

$$a_S = D_s A(|P_{e_s}|) + \max[F_s, 0] \quad (2.73d)$$

$$a_P = a_E + a_W + a_N + a_S + (F_e^u - F_w^u + F_n^u - F_s^u) + a_P^0 - S_p \Delta x \Delta y \quad (2.73e)$$

$$a_P^0 = \frac{\rho \Delta x \Delta y}{\Delta t} \quad (2.73f)$$

$$b = a_P^0 u_p^0 + S_c \Delta x \Delta y \quad (2.73g)$$

El término fuente se puede descomponer en:

$$\bar{S} \Delta x \Delta y = S_c + S_p u_p$$

La función de $A(|P_e|)$ es una función que depende del esquema numérico con el que se esté trabajando, como se muestra en la Tabla 2.3.

Análogamente se realiza la misma discretización para la ecuación de momentum y por lo tanto las ecuaciones de momentum (Ecs. (2.4 y 2.5)) discretizadas sobre su respectiva malla desplazada, se puede escribir en notación de coeficientes agrupados como:

$$a_{i,j}u_{i,j} = a_e^u u_e = \sum a_{vecinos}^u u_{vecinos} - \Delta y(P_E - P_P) + b^u \quad (2.74)$$

$$a_{l,j}u_{l,j} = a_w^v v_n = \sum a_{vecinos}^v v_{vecinos} - \Delta x(P_N - P_P) + b^v \quad (2.75)$$

Las ecuaciones discretizadas de conservación de momentum pueden ser resueltas si el campo de presión es conocido o estimado, P^* . El campo de velocidades obtenido puede no satisfacer la ecuación de continuidad a menos que la distribución de P^* sea el correcto. Para representar el nuevo campo de velocidades como u^* y v^* las ecuaciones anteriores (2.74 y 2.75) se pueden reescribir como:

$$a_e^u u_e^* = \sum a_{vecinos}^u u_{vecinos}^* - \Delta y(P_E^* - P_P^*) + b^u \quad (2.76)$$

$$a_n^v v_n^* = \sum a_{vecinos}^v v_{vecinos}^* - \Delta x(P_N^* - P_P^*) + b^v \quad (2.77)$$

Para la solución de las velocidades u^* y v^* en la ecuación de momentum, se requieren conocer los $F's$ y los $D's$ sobre sus respectivas mallas desplazadas. Debido al arreglo de la malla se requiere hacer una interpolación según sea el caso.

El siguiente paso a seguir es proponer, que la distribución correcta " P " se obtenga a partir de una corrección de presión " P' " y análogamente para las velocidades:

$$P = P^* + P' \quad (2.78)$$

$$u = u^* + u' \quad (2.79)$$

$$v = v^* + v' \quad (2.80)$$

Si a las ecuaciones de velocidades correctas Ecs. (2.74 y 2.75) se les restan las de velocidades supuestas Ecs. (2.76 Y 2.77), se obtiene una nueva ecuación de momentum para las correcciones de velocidades (Ecs. (2.79) y (2.80)) en función del campo de presión corregido (el término fuente se elimina ya que es el mismo para ambas ecuaciones):

$$a_e^u u_e' = \sum a_{vecinos}^u u_{vecinos}' - \Delta y (P_E' - P_P') \quad (2.81)$$

$$a_n^v v_n' = \sum a_{vecinos}^v v_{vecinos}' - \Delta x (P_N' - P_P') \quad (2.82)$$

Se introduce la aproximación de desvanecer los términos $\sum a_{vecinos}^u u_{vecinos}'$ y $\sum a_{vecinos}^v v_{vecinos}'$ con el objetivo de simplificar la relación entre las velocidades de corrección y la presión de corrección. La omisión de estos términos es la principal aproximación del algoritmo SIMPLE (Patankar, 1980), por lo que, las ecuaciones anteriores se pueden reducir a,

$$u_e' = d_e^u (P_P' - P_E') \quad (2.83)$$

$$v_n' = d_n^v (P_P' - P_N') \quad (2.84)$$

donde, $d_e^u = \frac{\Delta y}{a_e^u}$ y $d_n^v = \frac{\Delta x}{a_n^v}$; son los coeficientes que representan la relación entre las velocidades de corrección y la presión de corrección.

Conociendo las velocidades de corrección se pueden calcular las velocidades correctas a partir de las relaciones siguientes,

$$u_{i,j} = u_e = u_e^* + d_e^u (P_P' - P_E') \quad (2.85)$$

$$v_{l,j} = v_n = v_n^* + d_n^v (P_P' - P_N') \quad (2.86)$$

Como último paso se determina la información adecuada para la corrección de presión "P". Esta información se obtiene a partir de la ecuación de continuidad, la cual se integra sobre el VC de la malla principal o centrada, esto es:

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \quad (2.87)$$

Integrando la ecuación,

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho)}{\partial t} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u)}{\partial x} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho v)}{\partial y} dx dy dt = 0 \quad (2.88)$$

$$\int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho)}{\partial t} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho u)}{\partial x} dx dy dt + \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial(\rho v)}{\partial y} dx dy dt = 0 \quad (2.89)$$

resolviendo las integrales,

$$\frac{(\rho - \rho_P^0)}{\Delta t} \Delta x \Delta y + [(\rho u A)_e - (\rho u A)_w] + [(\rho v A)_n - (\rho v A)_s] = 0 \quad (2.90)$$

Sustituyendo la ecuaciones (2.83 y 2.84) en la ecuación anterior,

$$\begin{aligned} & \frac{(\rho - \rho_P^0)}{\Delta t} \Delta x \Delta y + [(\rho_e(u_e^* + d_e^u(P'_P - P'_E))) - (\rho_w(u_w^* + d_w^u(P'_W - P'_P)))] \Delta y + \\ & [(\rho_n(v_n^* + d_n^v(P'_P - P'_N))) - (\rho_s(v_s^* + d_s^v(P'_S - P'_P)))] \Delta x = 0 \end{aligned} \quad (2.91)$$

desarrollando términos,

$$\begin{aligned} & \frac{(\rho - \rho_P^0)}{\Delta t} \Delta x \Delta y + (\rho_e u_e^* + \rho_e d_e^u P'_P - \rho_e d_e^u P'_E - \rho_w u_w^* - \rho_w d_w^u P'_W + \rho_w d_w^u P'_P) \Delta y + \\ & (\rho_n v_n^* + \rho_n d_n^v P'_P - \rho_n d_n^v P'_N - \rho_s v_s^* - \rho_s d_s^v P'_S + \rho_s d_s^v P'_P) \Delta x = 0 \end{aligned}$$

agrupando términos,

$$\begin{aligned} & [\rho_e d_e^u \Delta y + \rho_w d_w^u \Delta y + \rho_n d_n^v \Delta x + \rho_s d_s^v \Delta x] P'_P = \rho_e d_e^u \Delta y P'_E + \rho_w d_w^u \Delta y P'_W + \\ & \rho_n d_n^v \Delta x P'_N + \rho_s d_s^v \Delta x P'_S - \rho_e u_e^* \Delta y + \rho_w u_w^* \Delta y - \rho_n v_n^* \Delta x + \rho_s v_s^* \Delta x - \frac{(\rho - \rho_P^0)}{\Delta t} \Delta x \Delta y \end{aligned}$$

Finalmente la ecuación anterior se expresa en notación de coeficientes agrupados:

$$a_P P'_P = a_E P'_E + a_W P'_W + a_N P'_N + a_S P'_S + b' \quad (2.92)$$

donde,

$$a_E = \rho_e d_e^u \Delta y$$

$$a_W = \rho_w d_w^u \Delta y$$

$$a_N = \rho_n d_n^v \Delta x$$

$$a_S = \rho_s d_s^v \Delta x$$

$$a_P = a_E + a_W + a_N + a_S$$

$$b' = \frac{(\rho - \rho_0)}{\Delta t} \Delta x \Delta y + [(\rho_w u_w^*) - (\rho_e u_e^*)] \Delta y + [(\rho_s v_s^*) - (\rho_n v_n^*)] \Delta x$$

Con esto se formulan las relaciones para determinar el campo de velocidades y el campo de presión al discretizar las ecuaciones de conservación de momentum.

Con el algoritmo SIMPLE se determina el campo de presión con el de velocidades aun existiendo el acoplamiento de las ecuaciones de continuidad con las de conservación de momentum, ya que es un método de solución secuencial para resolver el sistema de ecuaciones diferenciales parciales constituidas por la ecuación de continuidad y de conservación de momentum. Este consiste principalmente en suponer un valor para el campo de presión y con esto obtener las componentes del vector velocidad para después corregir dicho valor supuesto y así estimar un campo de velocidades corregido, todo esto se realiza iterativamente terminando al satisfacer la ecuación de continuidad por las componentes de velocidad, Patankar (1980) a continuación se enlista los pasos a seguir para la aplicación del algoritmo SIMPLE:

- 1.- Suponer un campo de presión y velocidades: P^* , u_{guess} y v_{guess} .
- 2.- Resolver las ecuaciones discretizadas de momentum para obtener: u^* y v^*
- 3.- Resolver la ecuación de corrección de presión: P'
- 4.- Calcular la presión P a partir de $P = P^* + P'$

5.- Calcular las componentes de velocidades con los valores de corrección de velocidades dadas por: $u = u^* + u'$ y $v = v^* + v'$

6.- Resolver otras variables.

7.- Aplicar el criterio de convergencia. Si se cumple el criterio, se imprimen los resultados y se concluye la solución numérica. En caso contrario, se continúa con el proceso iterativo.

8- Finalmente, en caso de continuar con el proceso iterativo renombrar $P^* = P$ y repetir desde el paso 2 hasta alcanzar convergencia y obtener el valor que cumpla con un criterio establecido.

Condiciones de frontera para la ecuación de presión corregida

La ecuación de la presión corregida P' no es una ecuación básica, por lo tanto, el tratamiento de sus condiciones de frontera deben ser comentadas. Hay dos clases de condiciones de frontera, con la presión conocida en la frontera (velocidad desconocida) ó con la componente de velocidad normal a la frontera dada.

Presión conocida en la frontera. Si el campo de presión propuesto P^* es arreglado de tal forma que en la frontera $P^* = P_{Frontera}$, entonces el valor de " P' " en la frontera debe ser cero.

Velocidad normal a la frontera dada. Si la malla es diseñada de tal forma que la frontera coincida con la cara del volumen de control, como se muestra en la Figura siguiente, en donde la velocidad u_n es dada. Por lo tanto, en la derivación de la ecuación de P' , para el volumen de control mostrado, no es necesario que el flujo a través de la frontera $(\rho v A)_n$ sea expresado en términos de v_n^* y su corrección correspondiente, pero si en términos de v_n . Entonces,

P'_N no aparecerá, ó a_N será cero en la ecuación de P' . Así que, ninguna información de P'_N será necesaria.

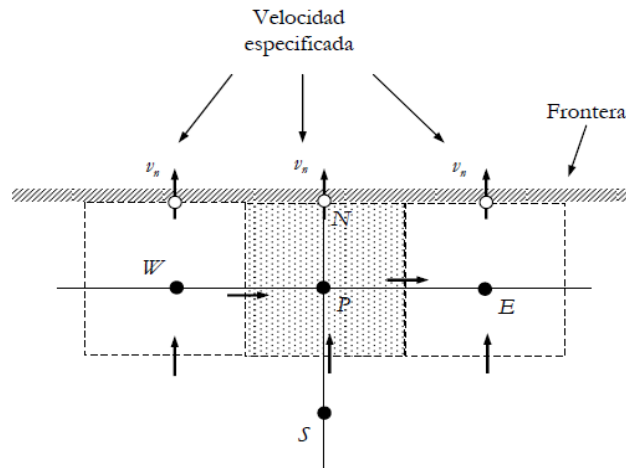


Figura 2.10 Volumen de control (VC) frontera para la ecuación de continuidad (J. Xamán et al., 2016).

2.2.4.3 Relajación de la solución parcial

Como se vio en el capítulo I, durante la solución numérica de problemas de flujos de fluidos, transferencia de calor y de masa se requiere de un proceso iterativo debido principalmente a la no-linealidad del problema, cuando se usa un método iterativo para la solución del sistema de ecuaciones algebraicas resultantes de método numérico de discretización, durante el proceso iterativo se puede acelerar o alentar los cambios de la variable ϕ entre una iteración y la siguiente. Por lo que es necesario, apoyarse del artificio numérico llamado bajo relajación como ya se explicó también en el capítulo I.

La bajo relajación es útil para los problemas que son no-lineales, ya que pueden aparecer oscilaciones que dificulten el proceso iterativo hacia la convergencia, por lo que, la bajo-relajación es a menudo usada para evitar la divergencia en el proceso iterativo de una solución numérica (Xáman, 2015), debido a que evita cambios bruscos de la variable.

Existen varias formas de introducir la bajo relajación a continuación se presentan dos formas:

A) RELAJACIÓN EN EL SOLVER

Partiendo de la ecuación de coeficientes agrupados para dos dimensiones ecuación (2.72).

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b$$

La cual también se puede escribir como:

$$a_P \phi_P = \sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b \quad (2.93)$$

Despejando ϕ_P ,

$$\phi_P = \frac{\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b}{a_P} \quad (2.94)$$

Sumando la variable de la iteración anterior ϕ_P^* , para no alterar la igualdad en ambos lados de la ecuación,

$$\phi_P + \phi_P^* = \phi_P^* + \frac{\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b}{a_P} \quad (2.95)$$

$$\phi_P = \phi_P^* + \left(\frac{\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b}{a_P} - \phi_P^* \right) \quad (2.96)$$

Se introduce el factor de relajación ($Fr = \alpha$) en la ecuación anterior, modificándose a,

$$\phi_P^{\text{Frelax}} = \phi_P^* + \alpha \left(\frac{\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b}{a_P} - \phi_P^* \right) \quad (2.97)$$

Con la introducción de factor de relajación en la ecuación anterior se está haciendo que ϕ_P^* cambie únicamente en una fracción *Frelax*. Se observa que

el primer término del paréntesis de ecuación (2.97) es ϕ_p sin relajar, entonces, si se sustituye este valor por la Ec. (2.94) se llega a,

$$\phi_p^{\text{Frelax}} = \phi_p^* + \alpha[\phi_p - \phi_p^*] \quad (2.98)$$

donde,

ϕ_p^{Frelax} es la variable relajada

ϕ_p^* es la variable de la iteración anterior

ϕ_p es la variable de la corriente iteración

B) RELAJACIÓN EN LOS COEFICIENTES

Partiendo de la ecuación (2.97)

$$\phi_p^{\text{Frelax}} = \phi_p^* + \alpha\left[\frac{\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b}{a_p} - \phi_p^*\right]$$

Se multiplica toda la ecuación por a_p ,

$$a_p \phi_p^{\text{Frelax}} = a_p \phi_p^* + \alpha[\sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b - a_p \phi_p^*] \quad (2.99)$$

Se divide toda la ecuación entre α (Fr),

$$\frac{a_p}{\alpha} \phi_p^{\text{Frelax}} = \frac{a_p}{\alpha} \phi_p^* + \sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b - a_p \phi_p^* \quad (2.100)$$

Factorizando de llega a,

$$\frac{a_p}{\alpha} \phi_p^{\text{Frelax}} = \sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b + \left[\frac{a_p}{\alpha} - a_p\right] \phi_p^* \quad (2.101)$$

$$\frac{a_p}{\alpha} \phi_p^{\text{Frelax}} = \sum_{\text{vecinos}} a_{\text{vecinos}} \phi_{\text{vecinos}} + b + \frac{a_p}{\alpha} [1 - \alpha] \phi_p^*$$

La ecuación anterior se puede escribir como ecuación en notación de coeficientes agrupados,

$$a_p^{Frelax} \phi_p^{Frelax} = \sum_{vecinos} a_{vecinos} \phi_{vecinos} + b^{Frelax} \quad (2.102)$$

donde, $a_p^{Frelax} = \frac{a_p}{Frelax};$ y $b^{Frelax} = b + \frac{a_p}{Frelax} [1-\alpha] \phi_p^*$

Como se mencionó también en el capítulo anterior el valor óptimo de α depende del caso bajo estudio a resolver. Un valor de $\alpha \approx 1$ permite acelerar el proceso iterativo y lo hace sensible a situaciones de divergencia; por el contrario, un valor de cercano a cero retarda el proceso iterativo y lo ayuda a evitar una divergencia. Aunque en general, es necesario emplear la experiencia y la intuición para elegir el valor más apropiado para cada caso, existen pocos estudios de investigación donde se ha empleado algún tipo de teoría para optimizar el valor de durante el proceso iterativo (J. Xamán et al., 2016), por lo que es un área de oportunidad para resolver el caso de estudio planteado con la metodología de la heurística de Recosido Simulado, la cual se explica a detalle en el capítulo siguiente.

CAPITULO III.METODOLOGÍA DE LA SOLUCIÓN

3.1 Metodología de la solución aplicada al algoritmo de recocido simulado

3.1.1 Introducción

Como se vio en el capítulo I un problema de optimización combinatoria consiste en encontrar entre un conjunto finito y discreto de soluciones (llamado espacio de búsqueda) la o las mejores soluciones que verifiquen un criterio particular.

Algunos algoritmos de optimización parten de una solución inicial que se supone es una mala solución y van encontrando iterativamente soluciones mejores que la anterior, aproximándose poco a poco a la solución óptima, y deteniéndose cuando ya no encuentran ninguna solución mejor, como se muestra en la Figura 3.1.

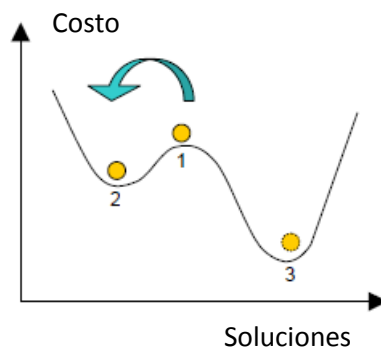


Figura 3.1 Ejemplo de un mínimo local

Como se muestra en la Figura anterior se supone que el algoritmo de optimización parte inicialmente de la posición 1, y opta por dirigirse hacia la izquierda llegando a la posición 2. Si el algoritmo es simple y sólo busca soluciones mejores no sabrá salir del 2, éste es un *mínimo local*, el mínimo global está en la posición 3, a la que nunca llegará, sin embargo existen algoritmos de optimización diseñados específicamente para evitar los mínimos locales y encontrar los mínimos globales.

Los algoritmos que se utilizan para resolver problemas de optimización combinatoria y que tratan de encontrar buenas soluciones en tiempos razonables se denominan algoritmos no-exactos, aproximados o heurísticos, (Nemhauser, L.A. Wosley, 1988) y (E-G. Talbi 2009).

Una pregunta natural que surge acerca de estos algoritmos de aproximación es cómo evaluar la calidad de las soluciones que se obtienen con éstos. En algunos casos, en los que se conoce la solución óptima del problema, basta con calcular el error relativo respecto de dicha solución óptima, sin embargo, muchas veces el valor de la solución óptima no es conocido, como es el caso de esta investigación, por lo que hay que aplicar otros métodos de validación estadística para determinar la calidad relativa de una nueva solución generada.

Debido a la complejidad inherente en muchos de los problemas de optimización combinatoria, se requiere de manera prácticamente obligatoria las estrategias de aproximación, aun cuando estos métodos sacrifican la garantía de encontrar una solución óptima a cambio de encontrar una solución válida en un tiempo razonable. Pues se garantiza que la solución hallada será habitualmente de alta calidad y se obtendrá mucho más rápidamente que en el caso de los algoritmos exactos. En la Figura 3.2 se muestra una clasificación de algunas técnicas de aproximación. No obstante, esta tesis se centra en el uso de una técnica metaheurística como estrategia de solución.

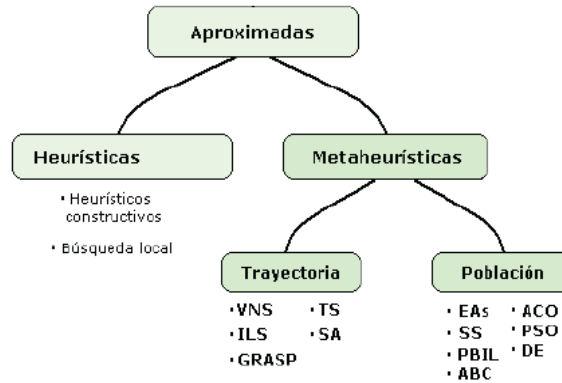


Figura 3.2 Clasificación de algunas de las técnicas de aproximación.

Desde los años setenta surgió un nuevo tipo de algoritmos aproximados, cuya idea central era combinar diferentes métodos heurísticos a un nivel más alto de manera que se lograra una exploración del espacio de búsqueda eficiente y efectiva, Estas técnicas se denominan metaheurísticas. El término fue introducido por primera vez por Glover en la década de los ochenta (antes de la aceptación de este término, a estas técnicas se les llamaba heurísticas modernas). Como es el caso de la metaheurística de recocido simulado del cual se habla a detalle en la siguiente sección de este capítulo.

3.1.2 Técnica metaheurística SA

Recocido simulado es una de las metaheurísticas más utilizadas en optimización combinatoria (Aarts y Lenstra, 2003). Está basada en el “principio de empeoramiento”, es decir, se permiten movimientos que empeoran la función objetivo (movimientos ascendentes) para escapar de óptimos locales. La técnica metaheurística de recocido simulado (*RS*), en inglés Simulated Anneling (*SA*), está basada en un algoritmo propuesto por Metrópolis et al. (1953) en el marco de la termodinámica estadística, para simular el proceso de enfriamiento de un material.

Este procedimiento se basa en una analogía con el comportamiento de un sistema físico del proceso de templado de metales el cual se utiliza en la industria para obtener materiales más resistentes, es decir, para mejorar las propiedades de un material, en otras palabras es una técnica para modificar el estado de un material y alcanzar un estado óptimo mediante el control de la temperatura. Inicia con el calentamiento del material a una temperatura alta, para luego enfriarlo lentamente, manteniendo en cada etapa una temperatura por cierto tiempo y con ello lograr la mejor configuración física. Lo anterior se ilustra en la figura siguiente.

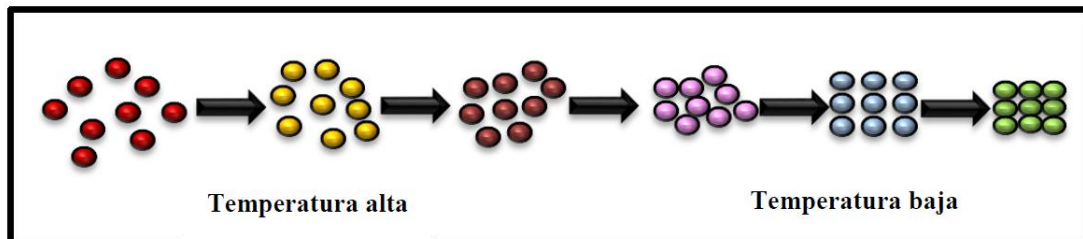


Figura 3.3 Configuraciones de un sólido aplicando recocido simulado.

El proceso de templado de metales consiste de tres fases: una fase de calentamiento a una temperatura determinada preferentemente **alta**, en esa situación los átomos adquieren una distribución no uniforme; en la segunda fase se sostiene una temperatura relativamente alta, lo cual, permite a las moléculas acomodarse en estados de mínima energía; y se sigue de una fase de enfriamiento controlado permitiendo que los átomos alcancen una condición de equilibrio (es decir, que los átomos alcancen una configuración óptima para esa temperatura). Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza así una máxima resistencia mientras que la energía del sistema es mínima.

Experimentalmente se comprueba que si la temperatura se hace descender bruscamente o no se espera suficiente tiempo en cada etapa, al final la

estructura del material se considera como no óptima. Por lo tanto, se puede considerar que la resistencia de la estructura depende de la velocidad de enfriamiento de los metales. Si, la temperatura inicial T_0 no es suficientemente alta o se aplica un enfriamiento rápido se obtienen imperfecciones (estados metaestables). En este caso, el sólido enfriado no alcanzará el equilibrio térmico en cada decremento de dicho parámetro. Los cristales fuertes se forman con un enfriamiento cuidadoso y lento. El algoritmo de recocido simulado aparenta los cambios de energía en un sistema sometido a un proceso de enfriamiento hasta que converge a un estado de equilibrio (estado congelado). Realiza la simulación de todo lo descrito anteriormente con respecto al recocido de los sólidos.

Kirkpatrick, Gelatt y Vecchi (1983) consideraron formular análogamente este proceso de recocido en algunos de los problemas de optimización combinatoria. Los autores consideraron que era posible establecer una analogía entre los parámetros que intervienen en la simulación termodinámica de Metrópolis y los que aparecen en los métodos de optimización, es decir, el uso del recocido simulado en optimización combinatoria se basa en establecer analogías entre el sistema físico y el problema de optimización. La Tabla 3.1 muestra las analogías entre ambas disciplinas y la analogía para el caso de prueba.

Tabla 3.1. Analogía entre proceso de recocido simulado en metalurgia, recocido simulado en optimización combinatoria y los casos de prueba

Metalurgia	Optimización combinatoria	Caso de prueba
Configuración	Solución factible	Que cumpla con el modelo de satisfacción de restricciones del problema
Energía de la configuración	Costo de la solución	Tiempo de convergencia en la solución del modelo matemático del problema
Mínima energía	Función objetivo	Mínimo tiempo de convergencia basado en la ecuación discreta de coeficientes agrupados y factores de relajación.
Configuración fundamental	Solución óptima	Factores de relajación que obtienen el mínimo valor de t_{conv}
Temperatura	Parámetro de control T_{SA}	Parámetro de control T_{SA}
Alcanzar el equilibrio	Longitud de la cadena de Markov (LCM), en cada ciclo de Metrópolis	Tamaño de la vecindad que define al número de posibles soluciones
Decremento de la temperatura	Coefficiente de control	Coefficiente de control que decremento a T_{SA}
Temperatura final	Criterio de paro	Criterio de paro, valor mínimo que alcanza T_{SA}
Metaestable	Óptimo local	Solución óptima local obtenida con los factores de relajación
Estado estable	Óptimo global (solución óptima)	Solución óptima global (mejor solución) obtenida con los factores optimizados

Para ello, se establece un paralelismo entre el proceso de las moléculas de una sustancia que van colocándose en diferentes niveles energéticos buscando un equilibrio, y las soluciones visitadas por medio de la técnica de búsqueda local. Por este motivo, se considera que SA es una metaheurística basada en búsqueda local en donde todo movimiento de mejora es aceptado y se permiten también movimientos de no mejora de acuerdo con algunas probabilidades. Las cuales están basadas en la analogía del proceso físico de enfriamiento y se obtienen como función del parámetro de control de la temperatura del sistema.

La forma de trabajar de la metaheurística es la siguiente:

Algoritmo de Metrópolis:

En el algoritmo de Metrópolis se genera una modificación en el sistema (generar un vecino) a partir de un estado de energía actual (solución actual), se genera un nuevo estado (modificación), muy parecido al actual y se calculan los cambios de energía resultantes (función de costo); si esta modificación (vecino), origina una disminución energética, se acepta la modificación, y si ocurre un incremento de energía, el cambio será aceptado con una probabilidad dada. A continuación se presentan los pasos anteriores.

- 1) El algoritmo comienza su ejecución con el parámetro de control de temperatura de recocido simulado el cual se denota por T_{SA} , con una *temperatura inicial* $T_{SA,0}$ **alta**.
- 2) Parte de una solución factible llamada S_{actual} , la cual puede ser generada en forma aleatoria, se evalúa la función de costo de esta solución (función objetivo), que se *identificará como* $f(S_{actual})$.

- 3) Mediante el mecanismo de perturbación, se genera una nueva solución S que corresponde a una solución vecina de S_{actual} , ésta solución vecina es S_{vecina} .
- 4) Se calcula el valor de la función de costo de $f(S_{vecina})$.
- 5) Se obtiene la diferencia de la función de costo entre ambas soluciones (ΔE) mediante la siguiente ecuación, la cual muestra la diferencia de energía:

$$\Delta E = f(S_{vecina}) - f(S_{actual}) \quad (3.1)$$
- 6) Se evalúa si, $\Delta E < 0$ indica que el costo de la nueva solución S_{vecina} es menor al costo de la solución actual S_{actual} , la nueva solución es aceptada, es decir, una solución de menor costo siempre se acepta.

Criterio de aceptación de Boltzmann:

En caso contrario, la nueva solución es aceptada con una probabilidad $P(\Delta E)$, definida por la siguiente ecuación llamada probabilidad de aceptación de Boltzmann:

$$P(\Delta E) = e^{-(\Delta E/T_{SA})} \quad (3.2)$$

A la ecuación (3.2), se le ha quitado la constante de Boltzman (k), ya que, ésta no se considera en los problemas de optimización combinatoria por no tener un significado dentro de la formulación del problema (Díaz y Dowsland, 2003).

Para una temperatura alta la probabilidad $P(\Delta E) \approx 1$, por lo que la mayoría de los cambios generados en el sistema son aceptados, es decir, que se aceptan soluciones de menor calidad. El aceptar una solución de menor calidad permite salir de un posible mínimo local y explorar otras áreas del espacio de

soluciones. Como la simulación comienza con una temperatura T_{SA} , **alta**, la $P(\Delta E) \approx 1$ y por lo tanto, una nueva solución con un costo mayor tiene una alta probabilidad de ser aceptada, conforme avanza la simulación el parámetro de temperatura va decreciendo y la $P(\Delta E) \approx 0$, de modo que, la mayoría de los cambios son rechazados (Dréo, Pétrowski, y Taillard, 2006) y (Díaz y Dowsland, 2003). Es decir, la probabilidad de aceptar una mala solución (que empeore el costo) va disminuyendo a medida que la T_{SA} decrece. Para cada nivel de temperatura, el sistema debe alcanzar un equilibrio, dicho de otra forma, un número de nuevas soluciones debe ser ensayado antes de que la temperatura sea reducida.

Este proceso se repite n veces según lo requiera el problema a resolver, cuando este ciclo iterativo se completa, T_{SA} se disminuye y comienza nuevamente el bucle interno, repitiendo el proceso de creación, evaluación y posible aceptación de soluciones vecinas. Cuando la temperatura de recocido simulado es lo suficientemente baja ($T_{SA,f} = \text{criterio de paro}$) el algoritmo finaliza y con ello se determina cuál fue la mejor solución obtenida.

El pseudocódigo del algoritmo básico de recocido simulado para minimización se presenta en la Figura 3.4.

Seleccionar parámetros de control

Repetir

Repetir (para cierta T_{SA})

Seleccionar aleatoriamente $S_{vecina} \in N(S_{actual})$

Sea $\Delta E = f(S_{vecina}) - f(S_{actual})$

Sea $\Delta E < 0$ entonces $S_{actual} = S_{vecina}$

En otro caso generar aleatoriamente $l \in U(0,1)$

Si $l < e^{-(\Delta E/T_{SA})}$ entonces $S_{actual} = S_{vecina}$

Hasta llegar al equilibrio

Actualizar el parámetro de control $T_{SA} = \alpha(T_{SA})$

Hasta condición de paro

Figura 3.4. Pseudocódigo del algoritmo de recocido simulado

Para la implementación de recocido simulado en la solución de un problema concreto es necesario tomar ciertas decisiones referentes a la definición de parámetros de control (temperatura de recocido simulado inicial $T_{SA,0}$, número de iteraciones en cada nivel de temperatura n , temperatura mínima $T_{SA,f}$), la tasa de decremento (α) de la temperatura de recocido simulado, el mecanismo para generar soluciones vecinas, entre otras. Estas decisiones se pueden clasificar en decisiones genéricas y específicas. Una transición consiste en la aplicación del mecanismo de generación de configuraciones vecinas y la aplicación del mecanismo de aceptación.

Para aplicar el algoritmo se requieren especificar 3 componentes:

- La representación del problema
 - ✚ Representar el espacio de solución
 - ✚ Expresar la función de costo también llamada función objetivo que represente adecuadamente el costo de cada solución.

- El mecanismo de transición para encontrar la mejor solución
 - ✚ Generar una nueva solución
 - ✚ Calcular diferencia de costo
 - ✚ Tomar decisión de aceptación

$$p(\text{aceptación}) = \begin{cases} 1 & \text{si } \Delta E < 0 \\ e^{-(\Delta E/T_{SA})} & \text{si } \Delta E \geq 0 \end{cases} \quad (3.3)$$

- El mecanismo de enfriamiento con el cual se llega a un estado estable

En la Figura 3.4, se pueden observar dos tipos de elementos que permiten afinar el algoritmo para que la ejecución del mismo sea la más eficaz de acuerdo con el problema que se desea resolver (Moreno Díaz et al., 2007):

Elementos genéricos: Son elementos que no tienen una dependencia directa del problema, sin embargo se deben sintonizar para saber cuáles son los valores que permiten la mejor representación del problema.

- Parámetro de control temperatura inicial del algoritmo de recocido simulado, $T_{SA,0}$

Como ya se mencionó el algoritmo debe comenzar con un valor de temperatura alto, que permita que muchos movimientos (o soluciones vecinas) sean aceptados. En la práctica, se requiere conocer el valor de la función de costo para las soluciones vecinas, por ejemplo, si el mayor incremento en la función objetivo (ΔE) entre soluciones vecinas es conocido, sería posible calcular un valor $T_{SA,0}$ que aceptase un movimiento con cierta probabilidad usando la ecuación 3.3. En la ausencia de tal conocimiento, se puede seleccionar una temperatura que parezca ser un valor alto, ejecutar el algoritmo para un tiempo corto y observar la tasa de aceptación. Si esta tasa es “convenientemente alta”, la temperatura con la cual se experimentó puede ser usada como el valor inicial $T_{SA,0}$. El significado de “convenientemente alta” varía de una situación a otra, pero en muchos casos una tasa de aceptación

ente 40% y 60% parece dar buenos resultados (Reeves, 1996) (Díaz & Dowsland, 2003).

- Proceso de enfriamiento $\alpha(T_{SA})$

Se refiere a la forma de la curva de enfriamiento que se utiliza en el recocido simulado, la cual determina la velocidad de disminución de la temperatura a medida que avanzan las iteraciones del algoritmo (Gendreau, Laporte, & Potvin, 2002). Existen diferentes maneras de abordar el decremento de la temperatura, una de las más utilizadas debido a su simplicidad y a los buenos resultados que ha dado en numerosas aplicaciones es la forma exponencial o geométrica, la ecuación de decremento de temperatura geométrica se muestra en la ecuación;

$$T_{k+1} = \alpha(T_k) \quad (3.4)$$

donde T_{k+1} es el parámetro de control de la temperatura en la iteración $k + 1$, T_k es el parámetro en la iteración k y α es el factor de decremento cercano a uno, escogido en el rango 0.85 a 0.99 (Kirkpatrick, Gelatt, y Vecchi, 1983).

- Número de repeticiones (iteraciones) n

El recocido simulado ejecuta cierto número de iteraciones en cada cambio del nivel del parámetro de control de temperatura para alcanzar el equilibrio, una vez alcanzado este estado la temperatura se reduce y el proceso se repite. Un esquema obvio es mantener un número de iteraciones constante en cada temperatura o alternativamente, se puede variar según desciende la temperatura, dedicando suficiente tiempo de búsqueda a temperaturas bajas para garantizar que se visita el óptimo local, es decir que conforme disminuye la temperatura se aumenta el número de repeticiones (Díaz y Dowsland, 2003).

- Condición de parada parámetro de control $T_{SA,f}$

Teóricamente la temperatura debería reducirse hasta 0, pero en la práctica la búsqueda converge por lo general a su óptimo local final antes de llegar a este valor de temperatura. Sin embargo, se puede correr el riesgo, de que el algoritmo se haga demasiado largo y los tiempos computacionales pueden ser muy grandes, por lo que el valor de este parámetro puede depender del problema en específico que se esté tratando y como en el caso del parámetro de control de temperatura inicial se puede hacer una estimación a partir de establecerla probabilidad de aceptación que se desea en la etapa final del algoritmo. Un criterio diferente es detener la búsqueda cuando se haya producido un número determinado de iteraciones sin alguna aceptación. Así por ejemplo, Dréo et al. (2006) sugieren que se debe terminar el algoritmo después de tres etapas sucesivas de temperatura sin que haya habido alguna aceptación.

Elementos dependientes del problema: Son los elementos que definen de forma directa el problema que se está resolviendo. Definen un modelo del problema y la estructura del espacio de soluciones para el mismo.

- Espacio de soluciones: s

Al conjunto de todas las posibles soluciones que pueden ser consideradas durante la ejecución del algoritmo para un problema concreto se llama espacio de búsqueda. El buscar una solución, se reduce a buscar un valor extremo (mínimo o máximo) en el espacio de búsqueda. A veces, este espacio de búsqueda puede ser bien definido, pero en la mayoría de las ocasiones sólo se conocen algunos puntos.

Existe un costo asociado a cada solución, de tal forma que el espacio de soluciones está caracterizado por un “paisaje de costo”. La dificultad de un problema de optimización radica en que este paisaje comprende un gran número de valles de profundidad variable y que corresponden a mínimos locales y de crestas de altura también variable y que se consideran óptimos

locales (ver Figura 3.5). La forma de este paisaje depende bastante de la función de costo y de la estructura de vecindad.

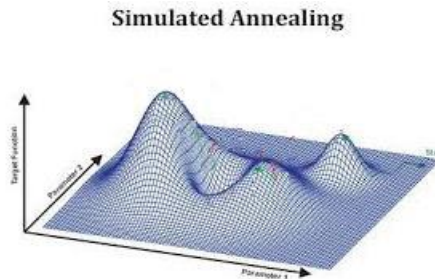


Figura 3.5. Representación de un espacio de soluciones aplicando el algoritmo de recocido simulado

- Estructura de vecindad: s'

En cada iteración, las configuraciones generadas de una solución inicial (s_0), definen un conjunto de soluciones vecinas de la solución s_0 , a esto se le llama vecindario $N(s_0)$ de s_0 . Una decisión importante en un problema de recocido simulado es la definición de la estructura de vecindad, o lo que es igual, establecer cómo generar una solución vecina. Los resultados teóricos demuestran que es suficiente con exigir que cualquier solución pueda alcanzarse desde cualquier otra a través de una serie de movimientos válidos denominados perturbaciones o movimientos (Reeves, 1996) y (Díaz & Dowsland, 2003).

- Función de costo: $f(s)$

Se define de tal manera que represente el problema que se está tratando de resolver. Un aspecto que es importante tomar en cuenta relacionado con la función de costo es el cálculo de su valor para una solución determinada. En cada iteración del algoritmo cuando se halla una nueva solución es necesario calcular el cambio de la función de costo con respecto a la solución anterior (Díaz & Dowsland, 2003).

- Solución inicial: s_0

Es un costo de la función objetivo puede ser factible o no, esta factibilidad depende del tipo de problema a resolver porque algunas metaheurísticas cuentan con mecanismos que las pueden hacer factibles. Se considera una solución factible porque cumple las restricciones requeridas por el problema a tratar.

3.2 Función de costo para optimizar los factores de relajación.

El objetivo principal para los casos de estudio planteados es minimizar el tiempo computacional, sin embargo, no siempre se puede obtener el menor tiempo posible aun cuando se consiga la convergencia, como se explicó en el capítulo II existe un artificio matemático llamado bajo-relajación que por medio de un factor o factores según sea el caso del problema a tratar, los cuales son llamados factores de relajación (*Frelax*), se pueden afectar directamente los cambios en la variable de estudio, que con la configuración adecuada de dichos factores se puede reducir de manera significativa el tiempo computacional requerido para dar solución al problema a resolver, la implementación matemática de la bajo relajación sobre el modelo matemático del caso 1 problema de convección-conducción y el caso 2 problema hidrodinámico se detalla en el capítulo anterior, la cual se toma como base para el planteamiento de la función objetivo a minimizar, como se muestra a continuación.

En base a la ecuación discretizada (ecuaciones 2.2 y 2.72) e incorporando los factores de relajación se llega a la ecuación 3.5, cabe mencionar que al discretizar el modelo matemático para cada caso planteado por el método de volumen finito se llega a la misma ecuación en notación de coeficientes

agrupados, sin embargo, la información que reciben los coeficientes de cada una de estas ecuaciones depende del caso de estudio a resolver (caso 1 y 2). Recordando que el objetivo central de este problema se basa en minimizar el tiempo de convergencia de la ecuación (3.5), en el cual se obtenga la convergencia del problema en el menor tiempo posible, la configuración adecuada de los valores de los factores de relajación afectan de manera trascendente para alcanzar la convergencia en el menor tiempo posible. SA por cada ciclo de metrópolis, obtiene la solución S_i (ecuación 3.5) de los problemas planteados (convección-conducción e hidrodinámico), con factores de relajación F_{relax} definidos con su estructura de vecindad, al resolver los problemas planteados se encuentra un tiempo de convergencia t_{conv} , definido por la ecuación (3.6). Esto se repite por cada ciclo de metrópolis de tal manera que se almacena siempre el mejor tiempo de convergencia y los factores de relajación que lo obtuvieron. Por lo que la función de costo a minimizar por SA es la que se presenta en la ecuación (3.7).

$$S_i = a_P^{F_{relax}} \phi_P - [a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S] + b^{F_{relax}} \quad (3.5)$$

$$t_{conv} = f(S_i, F_{relax}) \quad (3.6)$$

$$\min t_{conv} = \min f(S_i, F_{relax}) \quad (3.7)$$

3.3 Estrategia para optimizar los factores de relajación

Como ya se mencionó en este trabajo de tesis doctoral el algoritmo de recocido simulado ayuda a encontrar la mejor configuración para los valores de los diferentes factores de relajación que minimizan los tiempos de convergencia del problema. En el caso 1 se busca un factor de relajación F_{relax} para la variable de interés que es la temperatura T , mientras que para el caso 2 es

necesario un *Frelex* en cada una de las componentes de velocidad u y v y un *Frelex* para la presión P , al tener la mejor configuración se logra acelerar la convergencia en el problema de estudio.

A continuación se enumeran los pasos a seguir para la implementación de la metodología en la solución al problema planteado (casos 1 y 2) por medio de recocido simulado:

1. Implementar recocido simulado con la función de costo presentada en la ecuación 9, que obtiene la solución del problema, con los factores de relajación optimizados que reducen el tiempo de ejecución en la solución del problema en estudio.
2. Relajar el problema de estudio en la ejecución de recocido simulado: Para hacer un análisis experimental del comportamiento del problema hidrodinámico, se realizan diferentes pruebas con la implementación del algoritmo de recocido simulado, con las cuales se obtienen resultados parciales del problema con la finalidad de lograr que el método pueda ser funcional. Para obtener resultados parciales, se relajan las variables como tamaño de malla y el valor del residual (ε).
 - a) Relajación de la malla numérica. Para obtener resultados parciales se prueba con tamaños de malla más gruesa. Por ejemplo, si el tamaño de malla adecuado para este problema obtenido por un análisis de independencia de malla es de 61×61 , se relaja su valor con valores de malla gruesa de: 11×11 , 41×41 y 61×61 .
 - b) Relajación del valor óptimo del residual: Al igual que el tamaño de malla el valor del residual se relaja para lograr la funcionalidad del algoritmo. Por ejemplo, si el valor del residual adecuado para lograr la convergencia del problema es de $\varepsilon = 1 \times 10^{-10}$, se relaja su valor a un valor más grande $\varepsilon = 1 \times 10^{-03}$.

3. Una vez obtenidos los factores de relajación optimizados con recocido simulado del problema relajado, se prueban estos factores en el problema no relajado con tamaño de malla y residual adecuados.
4. Por último se hace un análisis y discusión de resultados

3.3.1 Estructura de vecindad para recocido simulado

Vecindad, se define como las posibles configuraciones generadas a partir de una solución inicial s del problema en estudio por medio de una pequeña perturbación, estas son el conjunto de soluciones vecinas a s , a esto se le llama vecindario o función de vecindad $N(s)$ de s . Una decisión importante en SA es la definición de la estructura de vecindad, es decir, la forma de establecer una nueva solución (tipo de perturbación). Los resultados teóricos demuestran que es suficiente con exigir que cualquier solución pueda ser alcanzada desde cualquier otra a través de una serie de movimientos válidos denominados perturbaciones o movimientos (Reeves, 1996; Díaz y Dowland, 2003). Para este trabajo la estructura de vecindad no realiza perturbaciones, lo que hace es elegir un decremento o incremento de ± 0.02 (valor sintonizado) en base al valor del intervalo válido de $0 < F_{relax} < 1$, por lo que la función de vecindad queda como $N(s, F_{relax})$. Para el caso 1 se busca optimizar el valor de F_{relaxT} para una sola variable de interés, mientras que, para el Caso 2 son tres los factores de relajación ($F_{relaxU}, F_{relaxV}, F_{relaxP}$), que se buscan optimizar a través de la meta heurística, puesto que son tres las variables de estudio. De forma aleatoria se asigna un valor inicial para cada uno de los factores, validando el intervalo dado, para la siguiente iteración se elige uno de los tres factores de forma aleatoria y se selecciona aleatoriamente en este un decremento o incremento de ± 0.02 (valor sintonizado), dejando los dos factores restantes con el valor anterior, y así sucesivamente para las demás iteraciones.

3.3.2 Algoritmo de recocido simulado (secuencial)

En el algoritmo de la Figura 3.6, se observa la implementación de SA para encontrar una configuración óptima de los factores de relajación para el problema en estudio; Inicialmente el algoritmo realiza una diversificación de la búsqueda con la aceptación de buenas y malas soluciones de forma indistinta, conforme avanza el proceso iterativo resulta cada vez más difícil el aceptar malas soluciones debido al criterio de aceptación que maneja SA.

En la línea 1, se define el valor del parámetro de control del algoritmo $T_{0,SA}$ el cual debe de ser inicializado con un valor alto, es decir debe ser mayor a cero para asegurar una probabilidad alta de aceptar un movimiento de no mejora, el valor de este parámetro se obtiene bajo una sintonización con la finalidad de identificar el valor correspondiente a dicha variable que permita influir de manera positiva en la calidad de la solución Martínez-Oropeza (2010). La línea 2, asigna $T_{SA} = T_{0,SA}$ donde T_{SA} es la variable que controla el ciclo externo de SA. La línea 3, selecciona el coeficiente de control de decremento α del parámetro de control T_{SA} , el cual determina el cambio que sufre en cada iteración T_{SA} , si el valor es muy pequeño reduce el tiempo de convergencia de SA y si es muy grande lo incrementa, este valor debe sintonizarse. La línea 4, selecciona la longitud de la cadena de *Markov*, la cual, permite al proceso alcanzar un estado Meta estable en cada valor del parámetro de control (T_{SA}), este parámetro define cuando se debe realizar un cambio de estado simulando un cambio de energía y representa el tamaño de la vecindad de una solución dada por el problema en estudio, por ejemplo de acuerdo a la estructura de vecindad que se maneja en el problema hidrodinámico, $LCM = 50$. La línea 5, muestra el criterio de paro representado por $T_{f,SA}$, el valor de este parámetro se obtiene mediante una sintonización. En la línea 6, se comienza con una solución inicial s_i del problema en estudio, que evalúa el costo de la función objetivo $f(s_i)$ (ver ecuación 3.6), la solución de inicio es de forma aleatoria, es decir, se fija un valor aleatorio para cada factor de relajación dentro del

intervalo $0 < F_{relax} < 1$. La línea 7, inicia el ciclo externo de SA, el cual hace un decremento en cada iteración del parámetro de control. La línea 8, evalúa el ciclo interno contabilizando la LCM , manteniendo a T_{SA} con un valor constante. La línea 9, obtiene una solución vecina s'_i , de acuerdo a la estructura de vecindad (ver sección 3.3.1). La línea 10, evalúa el costo de ambas soluciones obteniendo la diferencia en costo de la función objetivo (δ) en base a los tiempos de convergencia obtenidos del problema en estudio (ver sección 3.2).

Las líneas de la 11 a la 15 define el criterio de aceptación de una solución en SA, que puede ser mejor o peor que la que se tiene actualmente. La línea 11 evalúa si la solución vecina corresponde a una mejora en cuanto a la reducción en tiempo de convergencia del problema en estudio o si la solución vecina fue peor que la anterior. Si la solución vecina mejora el costo, esta se reemplaza como solución actual y se regresa nuevamente al ciclo interno para probar otra posible configuración vecina. La línea 12, indica que si la solución vecina fue peor que la actual se evalúa su aceptación por medio de la función de probabilidad de Boltzmann, la línea 13 genera un número aleatorio U dentro de un intervalo $[0,1)$, este número se compara en la línea 14 con el valor generado por Boltzmann, si U es menor que la probabilidad obtenida por Boltzmann, entonces la solución mala se acepta como nueva solución, este criterio de aceptación permite escapar de soluciones óptimas locales, lo que hace más eficiente a SA. La función de Boltzmann no considera de manera explícita la constante de Boltzmann, ya que, ésta no se considera en los problemas de optimización combinatoria por no tener un significado dentro de la formulación del problema Díaz et al., (2003). Es importante recalcar que la probabilidad de aceptación de s'_i en Boltzmann. En para valores altos de T_{SA} toma valores cercanos a uno, y, por lo tanto, la mayoría de las soluciones vecinas s'_i son aceptadas, en cambio, para valores bajos de T_{SA} la probabilidad de Boltzmann es cercana a cero, y en consecuencia la mayoría de

movimientos que no mejoran el costo de la función objetivo, son rechazados. La línea 15, termina la evaluación de la probabilidad de aceptación y se regresa a la línea 8 para continuar nuevamente con todo el proceso anteriormente descrito para el ciclo interno de SA. La línea 16, termina el ciclo interno una vez que se haya hecho toda la exploración de las posibles soluciones de acuerdo a la *LCM*. La línea 17, muestra el decremento de parámetro de control T_{SA} . La línea 18, evalúa el ciclo externo y se termina hasta que se cumpla el criterio de paro $T_{f,SA}$.

-
1. *Seleccionar un parámetro de control inicial $T_{0,SA} > 0$;*
 2. $T_{SA} = T_{0,SA}$;
 3. $\alpha =$ *velocidad de enfriamiento;*
 4. $LCM =$ *tamaño de vecindad;*
 5. $T_{f,SA} =$ *valor sintonizado;*
 6. $s_i =$ *solución_inicial(F_{relax});*
Sea $f(s_i)$ el costo de la solución s_i y $N(s_i, F_{relax})$ la función de vecindad;
 7. **REPETIR**
 8. **REPETIR**
 9. *Selecciona $s'_i \in N(s_i, F_{relax})$;*
 $s'_i =$ nuevos factores de relajación, vecinos de s_i
 10. *Sea $\delta = f(s'_i) - f(s_i)$;*
 11. *Si $\delta < 0$*
 ENTONCES $s_i = s'_i$;
 12. **SINO**
 13. *Genera aleatoriamente $u \in U[0,1)$;*
 14. *Si $u < e^{(-\frac{\delta}{T_{SA}})}$ ENTONCES $s_i = s'_i$;*
 15. **FINSINO**
 16. **HASTA QUE SE ACANCE LA LCM**
-

17. $T_{SA} = \alpha(T_{SA});$

18. HASTA QUE $T_{SA} \leq T_{f,SA}$

Figura 3.6. Recocido simulado implementado al problema de convección-conducción y al problema hidrodinámico.

3.4 Paralelización de algoritmos

En los últimos años proliferan cada vez más problemas que requieren un alto grado de cómputo para su resolución. La existencia de estos problemas complejos computacionalmente exige la fabricación de computadoras potentes para solucionarlos en un tiempo adecuado. Las computadoras secuenciales utilizan técnicas como segmentación o computación vectorial para incrementar su productividad; también se aprovechan de los avances tecnológicos, especialmente en la integración de circuitos, que les permite acelerar su velocidad de cómputo. Actualmente se han alcanzado límites físicos en el crecimiento de los beneficios, de modo que sobrepasarlos requerirá costes económicos no tolerables por los fabricantes y mucho menos por los usuarios. Es por ello que la programación paralela ha pasado una opción para tratar esta problemática.

La paralelización consiste, en que, en lugar de emplear el modelo estándar de computación utilizando un único recurso de procesamiento, se aplican técnicas de computación concurrente y paralelismo para abordar problemas complejos utilizando múltiples recursos de cómputo simultáneamente.

3.4.1 Técnicas de paralelización

Como ya se mencionó, *el paralelismo es la ejecución de varias actividades al mismo tiempo y que tienen una interrelación. El cómputo paralelo es la*

ejecución de más de un cálculo al mismo tiempo usando más de un procesador (García Regis y Cruz Martínez, 2003).

La técnica para la programación en paralelo, permite fragmentar un problema en subproblemas a resolver, su objetivo es dividir en forma equitativa los cálculos asociados al problema y los datos sobre los cuales opera el algoritmo y así reducir al mínimo el tiempo total de cómputo al distribuir la carga de trabajo entre varios procesadores. Las técnicas más difundidas son las de:

- ***Descomposición de dominio***, se concentra en el particionamiento de los datos del problema, intenta dividir los datos aproximadamente en el mismo tamaño para realizarles cálculos asociando las operaciones a datos específicos. Se asocia con la estrategia “divide y vencerás” y los modelos SIMD (Single Instruction Multi Data, memoria compartida) y SPMD (Single Program Multi Data, memoria distribuida). Un mismo programa puede ejecutarse sobre diferentes conjuntos de datos, pueden no realizar las mismas instrucciones. Es el modelo de la biblioteca MPI (Message Passing Interface).
- ***Descomposición funcional***, se concentra en el particionamiento de las operaciones del problema, replica los datos entre los procesos asociados a las diferentes tareas (realizan diferentes operaciones).

Cuando se comparte la carga de trabajo entre N procesadores, se espera que trabaje N veces más rápido que con un solo procesador pero no siempre resulta de esa forma, existen varios factores de sobrecarga que pueden disminuir el rendimiento. Algunos de estos factores son: las estructuras pocos regulares de los problemas, los mismos algoritmos o las técnicas utilizadas.

Una implementación adecuada en un sistema distribuido podría reducir el tiempo de cómputo dependiendo del número de procesadores usados en la paralelización para obtener un speedup mayor manteniendo la calidad del algoritmo secuencial.

3.4.2 Message Passing Interface (MPI)

Para poder paralelizar un algoritmo, es necesario contar con un lenguaje o biblioteca que permita hacerlo; dependiendo de esta herramienta y las restricciones del equipo de cómputo que se tenga, se “descompondrá” el cómputo secuencial en tareas para su ejecución en paralelo en varios procesadores, esto es conocido como “*granularidad*” (García Regis y Cruz Martínez, 2003).

Se le llama granularidad al tamaño de las tareas en que se divide un algoritmo, éstas pueden ser desde una sentencia de código, una función o un proceso que será ejecutado en paralelo.

- *Paralelización de grano fino*, cuando el código es dividido en tareas a nivel sentencia o cuando divide un ciclo en varios subciclos (se le conoce como Paralelismo de Instrucción).
- *Paralelización de grano medio*, el código se descompone a nivel subrutinas o segmentos de código (poca división del código).
- *Paralelización de grano grueso*, se le conoce como Paralelismo de Tareas. Es importante identificar las tareas del código que se pueden ejecutar en paralelo, esto es distribuir las operaciones en partes iguales entre la cantidad de procesadores a utilizar, sin olvidar que hay partes del código que deben ejecutarse de forma secuencial.

Ésta última es la que más se utiliza debido a su portabilidad y que se adapta a multiprocesadores de memoria compartida y distribuida. Se conocen tres estilos distintos de programación; paralelismo en datos, **programación por paso de mensajes (MPI)** y programación por paso de datos.

Las siglas **MPI** significan ‘Interfaz de Paso de Mensajes’, es una implementación de un estándar que define la sintaxis y la semántica de paso

de mensajes diseñados para ser utilizados en programas que exploten el paralelismo en arquitecturas de múltiples procesadores (Blaise Barney, 2014).

MPI puede definirse como: Una colección de rutinas que facilitan la comunicación entre procesadores en programas paralelos. Esta propuesta de ejecución del paralelismo mediante el envío de mensajes es a nivel de grano grueso y su principal esquema es la comunicación y generación de los procesos con el modelo Maestro-Esclavo (modelo que se verá en la siguiente sección). En este modelo, el maestro divide y distribuye el trabajo en pequeñas tareas que va asignando a cada nodo denominado como “esclavo”. Al terminar de ejecutar su tarea le envía los resultados de su trabajo al maestro para que éste recopile la información y se pueda mostrar al usuario.

MPI es un modelo de programación para aplicaciones en sistemas de memoria distribuida; una arquitectura de memoria distribuída está formada por un conjunto de procesadores, cada uno con su propia memoria local y espacio de direcciones. También es conocido como clúster, el cual se define como un conjunto de computadoras, también llamadas nodos, que están interconectadas mediante una red de alta velocidad que trabaja de manera conjunta para la solución de un problema, a través de configuraciones especiales que le habilitan la programación paralela a través del uso de bibliotecas de paso de mensajes. Cuando se ejecuta un algoritmo distribuido en un cluster, sus instrucciones y datos son distribuidos entre los procesadores con el objetivo de obtener un buen desempeño.

MPI ofrece varias características encaminadas a mejorar el rendimiento de los programas. Entre sus principales características destacan:

- Un conjunto de rutinas de comunicación punto a punto.
- Un grupo de rutinas de comunicación entre grupos de procesos.
- Un contexto de comunicación para el diseño de bibliotecas paralelas.
- La posibilidad de especificar diferentes topologías de comunicación.

- La posibilidad de crear tipos de datos derivados para enviar mensajes que contengan datos no contiguos en memoria.
- En MPI se puede hacer uso de comunicación asíncrona.
- MPI maneja más eficientemente el paso de mensajes.
- MPI es totalmente portable.
- Existen varias implementaciones de calidad de MPI disponibles (LAM, MPICH, CHIMP).
- Existe una implementación de MPI para C.
- Hay especificación formal de MPI.
- MPI es un estándar.

Cada programa en MPI debe incluir la librería `mpi.h` que contiene las definiciones y las declaraciones necesarias para compilar un programa que use la librería.

Todos los identificadores de MPI contienen el prefijo 'MPI' en sus nombres de funciones. Antes de llamar a cualquier función MPI, es necesario invocar además `MPI Init`. Cuando el programa termine de ejecutarse debe invocarse `MPI Finalize`.

Se puede resumir la estructura en 3 pasos básicos:

1) Inicializar la comunicación

MPI_Init ()	Función que inicializa el ambiente MPI
MPI_comm_size ()	Retorna el número de procesos
MPI_Comm_rank ()	Indica el número de cada procesador

2) Comunicar para compartir datos entre procesos (punto a punto)

MPI_Send ()	Envía un mensaje
MPI_Recv ()	Recibe un mensaje

3) Finalizar el ambiente paralelo

MPI_Finalize ()

Finaliza el ambiente paralelo de MPI

La estructura básica de un programa en MPI se muestra en la siguiente Figura 3.7.

```
1. # include <mpi.h>
2. int main() {
3. int nproc; /*Número de procesos*/
4. int rank; /*Mi dirección; 0<=rank<=(nproc-1)*/
5. MPI_Init(&argc, &argv);
6. MPI_Comm_size(MPI_COMM_WORLD,&nproc);
7. MPI_COMM_rank(MPI_COMM_WORLD,&rank);
8. /*CUERPO DEL PROGRAMA*/
9. MPI_Finalize();
10. Return 0;
11. }
```

Figura 3.7. Estructura básica de un algoritmo en MPI

3.4.3 Modelos de comunicación entre procesos

Estos modelos se utilizan para comunicar y/o sincronizar procesos paralelos. Los modelos de comunicación son:

- a) **Modelo maestro-esclavo.** En las redes de computadoras, este es un modelo para un protocolo de comunicación en el que un proceso (proceso maestro) impone la dirección del control del programa a uno o

más procesos (esclavos) que son los que procesan la información, la única comunicación que tienen con el maestro es la entrega de resultados de la tarea asignada. En la descomposición de dominio los esclavos son idénticos (homogéneos). En la descomposición funcional los esclavos son diferentes (heterogéneos). En la Figura 3.8 se muestra este modelo.

Las variantes que tiene es que pueden ser síncronos o asíncronos, el maestro puede procesar o no (activo o pasivo); el mecanismo de asignación de datos por parte del maestro puede ser dinámico o estático.

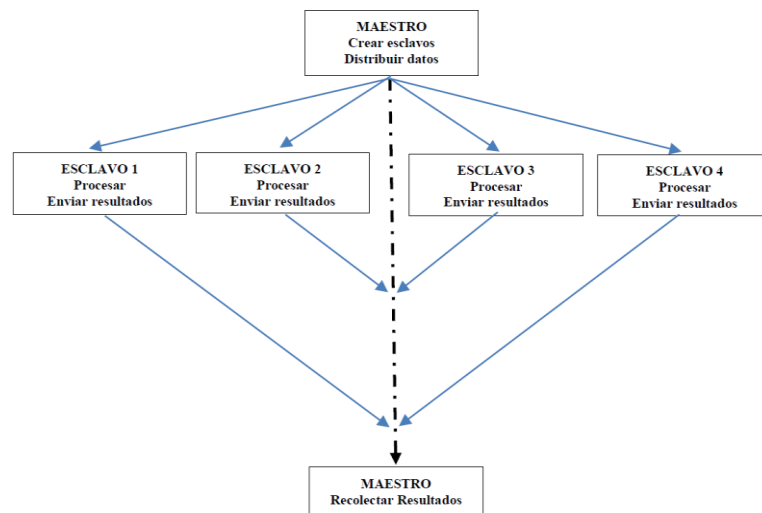


Figura 3.8 Modelo Maestro-Esclavo

- b) **Modelo cliente servidor.** Identifica dos clases de procesos: terminales (los clientes) solicitan servicios a procesos de otra clase (los servidores) que atienden los pedidos. El modelo proporciona un mecanismo para comunicar aplicaciones distribuidas (que funcionen simultáneamente como clientes y servidores para distintos servicios). Los clientes son los elementos que necesitan servicios del recurso y el servidor es la entidad

que posee el recurso. El proceso servidor está permanentemente activo aguardando solicitudes de parte del cliente. La Figura 3.9 ejemplifica este modelo.

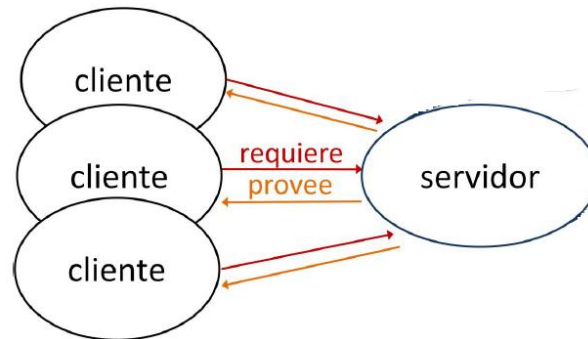


Figura 3.9 Modelo Cliente-Servidor

- c) **Modelo peer to peer.** Modelo de comunicación en el cual cada componente tiene las mismas capacidades para establecer la comunicación. Red de pares, sin clientes ni servidores fijos, se implementa asignando a los pares las capacidades de cliente y servidor, se aprovechan, administran y optimizan el uso del ancho de banda por medio de la conectividad entre usuarios, obteniendo más rendimiento en las conexiones y transferencias que los métodos centralizados convencionales.

3.4.4 Tipos de comunicación

Un punto importante en MPI es el tipo de comunicaciones, se determinan con base en el número de procesos que intervienen en la comunicación. Se conocen dos tipos de comunicación:

- 1) **Punto a punto**, es el mecanismo básico de transferencia de mensajes entre un par de procesos, uno enviando y el otro recibiendo. MPI provee de un conjunto de funciones de envío y recepción de mensajes que permiten la comunicación de datos de cierto tipo con una etiqueta

asociada (Hidrobo, 2005). La comunicación punto a punto consiste en tener exactamente un proceso emisor que ejecuta una operación de envío y un proceso receptor que ejecuta la operación de recepción, esto puede observarse en la Figura 3.10.

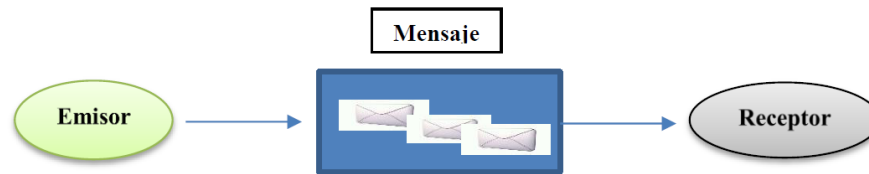


Figura 3.10. Comunicación punto a punto

Existen ciertas condiciones que deben tomarse en cuenta para poder trabajar con la comunicación punto a punto:

- Cada mensaje enviado dentro de la red de comunicación será recibido por el receptor procesador de forma secuencial.
- Debe de existir siempre un proceso receptor de los mensajes enviados, de no ser así, ocurrirán errores.
- Se debe cuidar que las variables enviadas sean del mismo tipo: enteros, flotantes, etc.

Dentro de la comunicación punto a punto se tienen dos tipos de comunicaciones:

a) Comunicación bloqueante, en este tipo de comunicación, el procesador emisor tiene que “esperar” hasta que el procesador receptor complete las etapas de su programa para que pueda continuar su ejecución. Tiene la desventaja de producir tiempos muertos que afectan el aprovechamiento y utilización de los procesadores.

b) Comunicación no bloqueante, su característica es que no interrumpe la ejecución del procesador cuando se realizan operaciones de envío y recepción.

2) **Colectiva**, se usan cuando se requiere enviar un conjunto de datos a múltiples procesos o cuando se necesita recibir en un único proceso datos de varios remitentes. Este tipo de comunicaciones permiten la transferencia de datos entre todos los procesos que pertenecen a un grupo específico, se usan comunicadores en lugar de etiquetas. En este tipo de comunicación todos los procesos que pertenecen a un grupo están involucrados en el intercambio de datos de manera simultánea.

En la figura 3.11 se muestra un ejemplo de comunicación colectiva que es la distribución de un dato de un emisor a varios receptores.

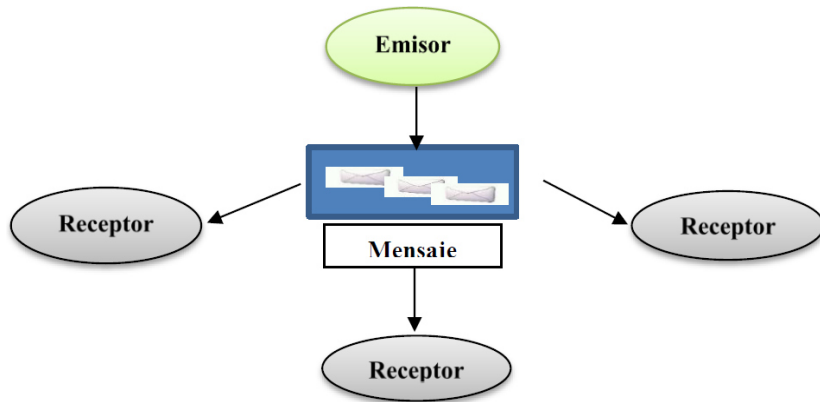


Figura 3.11. Comunicación colectiva

3.4.5 Recocido simulado (paralelo)

Para implementar la paralelización en este trabajo se considera el modelo maestro esclavo considerando que el mismo programa se ejecute en los diferentes procesadores (un solo programa con múltiples datos SPMD por sus siglas en inglés). En la Figura 3.12, se observa que solos se dan dos comunicaciones entre el nodo maestro y los esclavos, al inicio cuando el nodo maestro entrega la información y la segunda vez son los esclavos quienes se comunican con el maestro para entregarle los resultados.

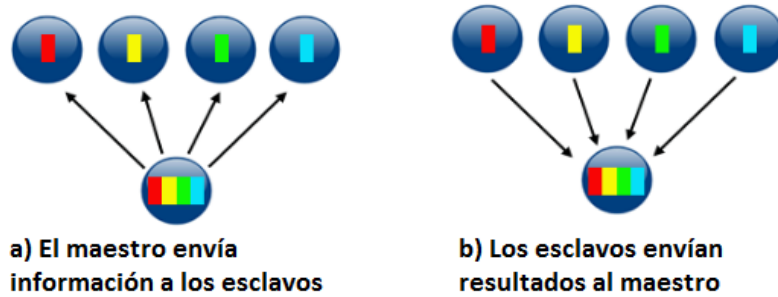


Figura 3.12 Momentos de comunicación entre maestro y esclavos

Cada procesador tiene asignada una configuración de la solución inicial y cada uno ejecuta el algoritmo de recocido simulado, en cada uno se obtendrán k soluciones independientes $S1N, S2N \dots SkN$, entre las cuales se selecciona la mejor de todas como solución final, la probabilidad de encontrar la solución óptima aumenta con el número de procesadores.

El modelo de islas es una implementación paralela propuesta por Pettey et al., 1987, la cual ha demostrado ser muy eficiente al ser aplicada en los algoritmos genéticos de acuerdo a la semejanza que hay con respecto a los seres vivos, sin embargo, se propone este modelo por la implementación realizada, pues en que cada procesador se ejecuta el algoritmo de recocido simulado y se encuentra una solución por cada uno de ellos, esto se muestra en la Figura 3.13.

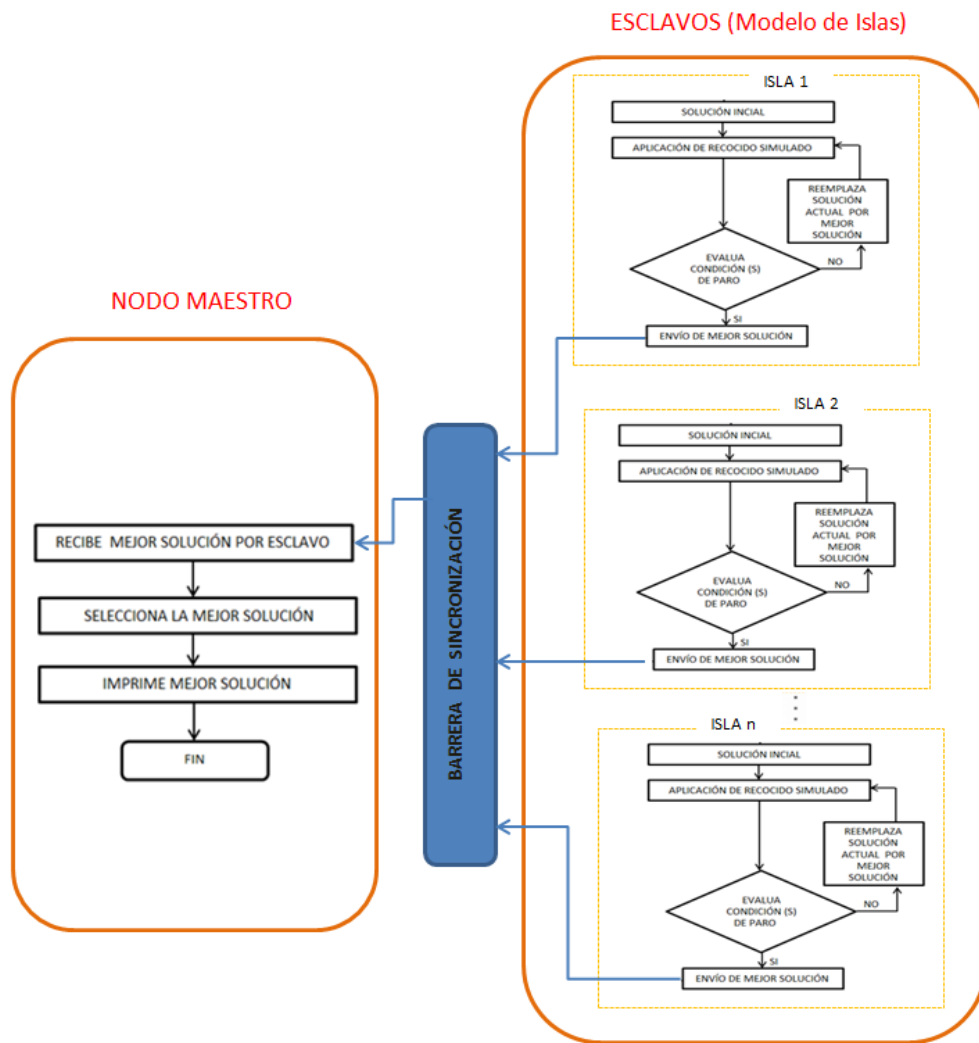


Figura 3.13. Diagrama del algoritmo de recocido simulado distribuido

CAPITULO IV. PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS

4.1 Resultados experimentales para el algoritmo secuencial

En este capítulo, se valida de manera experimental el desempeño del algoritmo de recocido simulado desarrollado en este trabajo y descrito en capítulos anteriores. Se muestra el desempeño en función de la eficiencia y la eficacia en el algoritmo secuencial y distribuido, además de la infraestructura utilizada en el desarrollo de esta investigación.

Para las pruebas experimentales de la programación secuencial se utilizó una laptop, ® core™ Processor i7-3612QM CPU @ 2.10 GHz, 8 GB RAM, Windows 7 64 bits, Visual C++ 2010.

Caso 1: problema de convección-conducción

La Tabla 4.1 muestra los valores sintonizados de los parámetros de control de recocido simulado, valores con los cuales se obtiene la convergencia del algoritmo para el caso de prueba 1.

Tabla 4.1. Valores sintonizados para los parámetros de control de recocido simulado

Parámetro de control	Valor
$T_{0,SA}$	1000
$T_{f,SA}$	10
LCM	40
α	0.965

En la Figura 4.1 se muestra de forma gráfica el análisis de sensibilidad para los valores mostrados en la Tabla 4.1, en esta figura se muestra la probabilidad de aceptación con respecto al número de iteraciones de SA, se observa que conforme comienza a iterar SA la probabilidad de aceptar malos resultados disminuye hasta alcanzar la convergencia del algoritmo de SA.

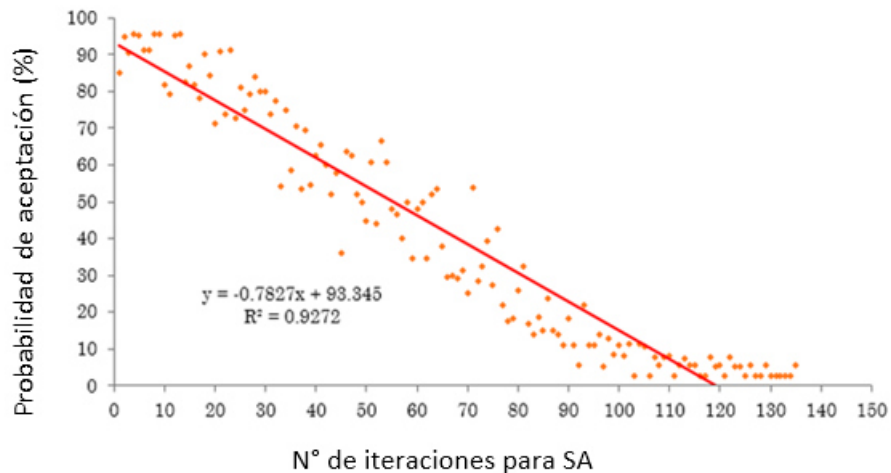


Figura 4.1 Análisis de sensibilidad para los parámetros de SA

Puesto que la metodología utilizada se trata de un método estocástico es necesario analizar los resultados optimizados de SA con un mínimo de 30 ejecuciones por cada problema en cada una de las pruebas realizadas con diferentes valores en malla y residual. La Tabla 4.2, muestra las treinta ejecuciones de la prueba con malla de 41×41 y un residual de $\epsilon = 1 \times 10^{-05}$, para el caso 1. Se muestra el número de ejecución que se realiza de SA, el factor de relajación optimizado y el tiempo de convergencia que se obtiene del problema de transferencia de calor. El rango obtenido en las 30 pruebas con respecto al tiempo de convergencia es de $t = 0.025$ con un mínimo de $t = 0.01$, un máximo de $t = 0.026$ y una media de 0.019 . La moda es de 0.018 y la desviación estándar es de 0.00435 . Se observa con esto, que los resultados en tiempo de convergencia de las 30 pruebas de SA para el problema de transferencia de calor, no se disparan a valores altos y estos presentan un buen comportamiento de acuerdo a la desviación estándar.

Tabla 4.2. Ejecuciones realizadas por SA para una malla de 41x41 y un residual de $\varepsilon = 1 \times 10^{-05}$, para el problema del caso 1.

N° de ejecución	$Frelax_T$	tiempo (s)
1	0.967	0.013
2	0.99	0.01
3	0.975	0.011
4	0.976	0.012
5	0.971	0.013
6	0.88	0.02
7	0.889	0.019
8	0.876	0.02
9	0.897	0.018
10	0.9	0.018
11	0.831	0.025
12	0.88	0.021
13	0.908	0.018
14	0.903	0.019
15	0.91	0.017
16	0.928	0.016
17	0.921	0.016
18	0.925	0.017
19	0.93	0.015
20	0.87	0.022
21	0.854	0.024
22	0.86	0.023
23	0.831	0.026
24	0.858	0.023
25	0.859	0.023
26	0.844	0.025
27	0.849	0.024
28	0.91	0.019
29	0.913	0.018
30	0.92	0.017

La Figura 4.2, presenta el comportamiento del tiempo computacional con respecto al factor de relación $Frelax_T$, con una aproximación de mínimos cuadrados, se observa que existe una relación inversamente proporcional, de manera general, cuando el factor de relajación del problema de transferencia de calor aumenta, entonces el tiempo de convergencia disminuye. Sin

embrago, también se puede observar que existen muchos puntos como los marcados en círculo rojo, que muestran que con un pequeño decremento del $Fre lax_T$, el tiempo de convergencia tiende a aumentar, esto es indicativo de que no necesariamente se sigue una tendencia directamente proporcional.

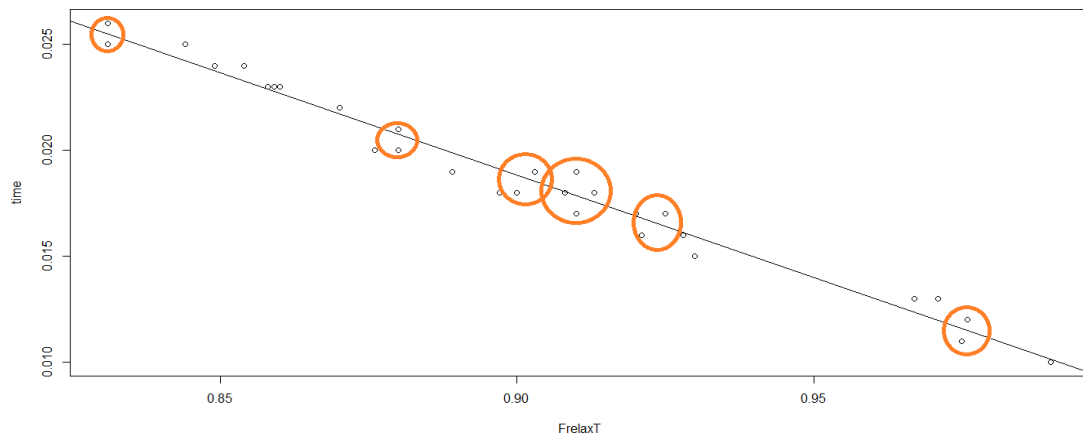


Figura 4.2. Comportamiento en el tiempo de convergencia del problema del caso 1 en función del factor de relajación.

Tabla 4.3, muestra los resultados de los tiempos de convergencia t , con diferente factor de relajación para la variable T del problema de transferencia de calor del caso 1. Se realizan tres pruebas para distintos tamaños de malla relajando el valor del residual $\epsilon=1 \times 10^{-05}$, se comparan los mejores valores encontrados por SA con respecto a valores propuestos de forma empírica. Se puede observar que para cada caso se mejora en más del 60% en el tiempo computacional utilizando los valores que proporciona la heurística, llegando arriba del 90% con una malla de 41×41 .

Tabla 4.3. Factores de relajación obtenidos con SA, con mallas y residual relajados ($\varepsilon = 1 \times 10^{-05}$), para el problema del caso 1.

Tamaño de malla	Factores de relajación propuestos	tiempo (s)	Factores de relajación encontrados con SA	tiempo (s)	% mejora en el tiempo
	$Frelax_T$		$Frelax_T$		
11x11	0.2	0.003	0.66	0.001	67
41x41	0.2	0.137	0.99	0.010	93
61x61	0.2	0.258	0.78	0.027	81

Comparando los resultados de convergencia obtenidos en la Tabla 4.2 por SA para el problema de transferencia de calor, con el resultado obtenido en la Tabla 4.3, se puede observar que el factor de relajación propuesto no supera el resultado obtenido por SA en ninguna de las 30 ejecuciones, cuando el tamaño de malla es de 41x41.

De las 30 pruebas realizadas por SA para una malla de 61x61 y $\varepsilon=1 \times 10^{-05}$, se toman los tres mejores valores de los factores de relajación encontrados que obtuvieron los mejores tiempos de convergencia para el caso 1 y se evalúan con el mismo tamaño de malla (no relajada) de 61x61 pero ahora con un residual no relajado $\varepsilon=1 \times 10^{-10}$. Los resultados se comparan con factores de relajación propuestos de forma empírica, los resultados se muestran en la Tabla 4.4, se observa que para la primera prueba se da una mejora en tiempo de convergencia del problema en un 60% mientras que para los dos casos restantes se mejora en más del 70%.

Tabla 4.4. Factores de relajación obtenidos con SA, con malla y residual ($\varepsilon = 1 \times 10^{-10}$) no relajados, para el problema del caso 1.

Tamaño de malla	Factores de relajación propuestos	tiempo (s)	Factores de relajación encontrados con SA	tiempo (s)	% mejora en el tiempo
	$Frelax_T$		$Frelax_T$		
61x61	0.4	0.208	0.66	0.084	60
61x61	0.4	0.208	0.88	0.034	83
61x61	0.4	0.208	0.78	0.054	74

Caso 2: problema hidrodinámico

Tabla 4.5 muestra los valores sintonizados de los parámetros de control de recocido simulado, valores con los cuales se obtiene la convergencia del algoritmo para el caso de prueba 2.

Tabla 4.5. Valores sintonizados para los parámetros de control de recocido simulado para el caso de prueba 2.

Parámetro de control	Valor
$T_{0,SA}$	1000
$T_{f,SA}$	10
LCM	50
α	0.95

Tabla 4.6 muestra las treinta ejecuciones de la prueba con malla de 11x11 y un residual de $\varepsilon=1 \times 10^{-05}$, para el caso 2. Se muestra el número de ejecución que se realiza de SA, el factor de relajación optimizado y el tiempo de convergencia que se obtiene del problema de transferencia de calor. El rango obtenido en las 30 pruebas con respecto al tiempo de convergencia es de $t = 0.016$ con un mínimo de $t = 0.023$, un máximo de $t = 0.039$ y una media de 0.032. La moda es de 0.035 y la desviación estándar es de 0.00438. Se

observa con esto, que los resultados en tiempo de convergencia de las 30 pruebas de SA para el problema hidrodinámico, no se disparan a valores altos y estos presentan un buen comportamiento de acuerdo a la desviación estándar.

Tabla 4.6. Ejecuciones realizadas para una malla de 11x11 un residual de $\varepsilon = 1 \times 10^{-05}$, para el problema del caso 2.

N° de ejecución	Factores de Relajación			tiempo (s)
	$Frelax_U$	$Frelax_V$	$Frelax_P$	
1	0.91	0.72	0.24	0.028
2	0.64	0.62	0.22	0.037
3	0.89	0.88	0.43	0.034
4	0.99	0.55	0.46	0.038
5	0.78	0.8	0.44	0.036
6	0.92	0.64	0.61	0.035
7	0.62	0.6	0.22	0.036
8	0.82	0.54	0.32	0.035
9	0.97	0.99	0.53	0.029
10	0.99	0.93	0.55	0.031
11	0.65	0.51	0.19	0.039
12	0.85	0.75	0.17	0.03
13	0.86	0.72	0.17	0.031
14	0.68	0.78	0.23	0.035
15	0.93	0.79	0.24	0.026
16	0.97	0.85	0.27	0.023
17	0.84	0.52	0.23	0.035
18	0.95	0.72	0.23	0.028
19	0.69	0.89	0.22	0.032
20	0.61	0.89	0.21	0.036
21	0.93	0.99	0.54	0.03
22	0.91	0.95	0.55	0.024
23	0.85	0.83	0.56	0.035
24	0.9	0.95	0.28	0.027
25	0.86	0.4	0.41	0.037
26	0.88	0.9	0.28	0.026
27	0.94	0.5	0.37	0.032
28	0.92	0.48	0.38	0.034
29	0.81	0.73	0.25	0.027
30	0.99	0.55	0.34	0.03

En Figura 4.3, muestra el comportamiento de los tiempos de convergencia para el problema hidrodinámico con respecto al número de ejecuciones del algoritmo de SA para las treinta pruebas realizadas, los resultados del tiempo de convergencia están en función de los valores de la mejor configuración optimizada de los factores de relajación, encontrados por SA y presentados en la tabla 4.6.

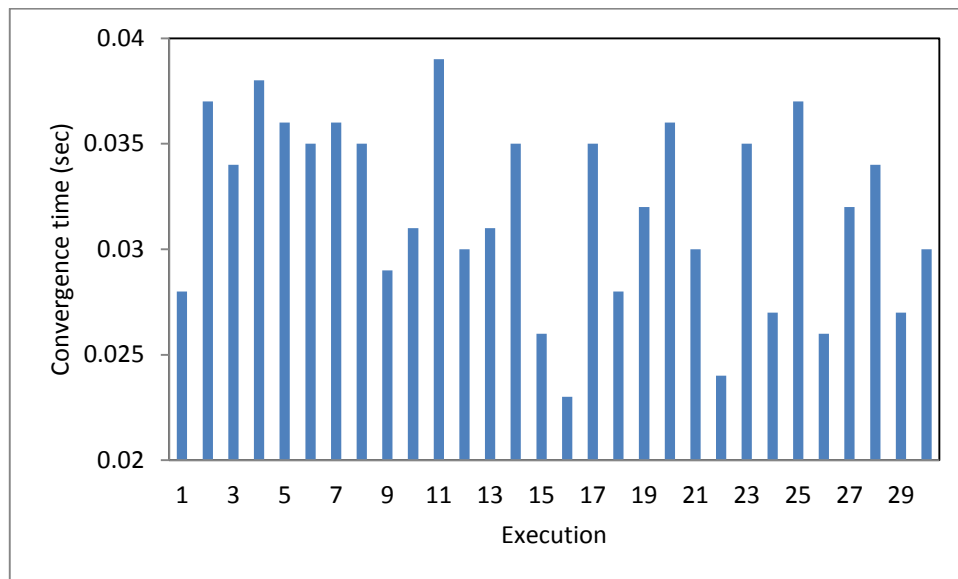


Figure 4.3. Algoritmo SA. Optimización del tiempo de convergencia del problema hidrodinámico, caso 2. Malla= 11x11, $\varepsilon = 1x10^{-05}$.

Figura 4.4 muestra el comportamiento de los factores de relajación presentados en la Tabla 4.4, para el problema hidrodinámico con respecto al número de ejecuciones del algoritmo de SA para las treinta pruebas realizadas, los resultados del tiempo de convergencia están en función de los valores de la mejor configuración optimizada de los factores de relajación. El tiempo de convergencia ($t \times 10^{-3}$) del problema hidrodinámico se presenta como una etiqueta en cada coordenada (*FrelaxU*, *FrelaxV*, *FrelaxP*). Aquí se observa que el tiempo de convergencia presenta una ligera tendencia a disminuir cuando los factores de relajación tienden a incrementar, pero este comportamiento no es totalmente proporcional como sucede también en el

problema del caso 1. En círculo rojo aparece el mejor tiempo de convergencia y en círculo azul aparece el peor tiempo de convergencia, sin embargo, existen tiempos de convergencia elevados con valores grandes en los factores de relajación, también se observa que el valor de convergencia (círculo amarillo) que más se repite (moda), se distribuye en la parte de factores de relajación con valores pequeños como en la parte de factores de relajación con valores grandes.

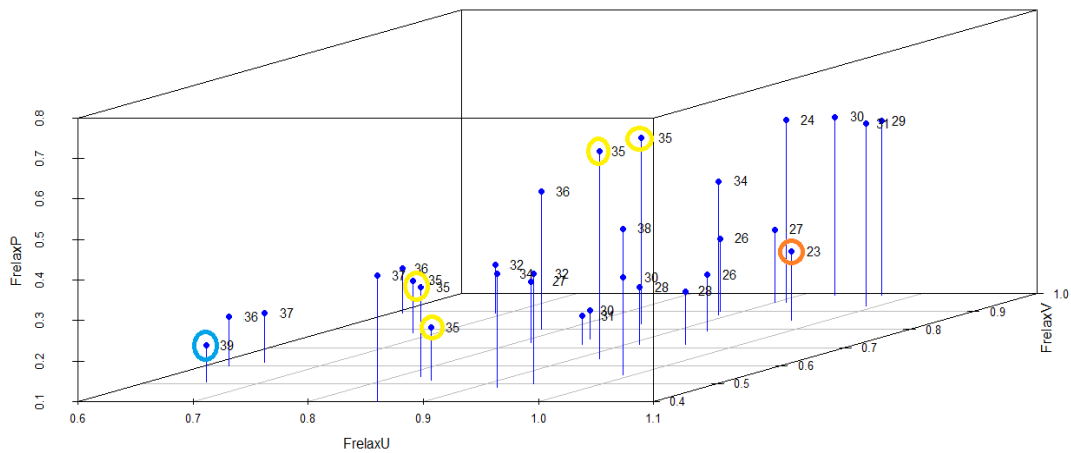


Figura 4.4. Factores de relajación para una malla de 11x11 y un residual de $\varepsilon = 1 \times 10^{-05}$, para el problema del caso 2.

La Tabla 4.7 muestra los resultados de convergencia en tiempos obtenidos con un $\varepsilon = 1 \times 10^{-03}$ y variando el tamaño de malla, la comparación se realiza con factores de relajación propuestos por simple deducción empírica o conocimiento previo acerca del tema y la mejor configuración encontrada de factores de relajación con SA.

Tabla 4.7. Factores de relajación obtenidos con SA, con diferentes tamaños de mallas y residual de ($\varepsilon = 1 \times 10^{-05}$) para el problema del caso 2.

Tamaño de malla	Factores de relajación propuestos			tiempo (s)	Factores de relajación encontrados con SA			tiempo (s)	% mejora en el tiempo
	$Frelax_U$	$Frelax_V$	$Frelax_P$		$Frelax_U$	$Frelax_V$	$Frelax_P$		
11x11	0.2	0.2	0.1	0.113	0.82	0.87	0.27	0.029	74
41x41	0.2	0.2	0.1	2.981	0.43	0.84	0.51	0.694	76
61x61	0.2	0.2	0.1	12.006	0.87	0.58	0.42	1.629	86

Se observa en la Tabla 4.7 que para cada uno de las pruebas realizadas 16variando el tamaño de la malla se mejora en más del 50% el tiempo computacional en la solución del problema hidrodinámico utilizando los factores encontrados con la metaheurística de recocido simulado en comparación con los factores propuestos. Comparando los resultados de convergencia obtenidos por SA para el problema de hidrodinámico (tabla 4.6) con el resultado obtenido en la Tabla 4.7, se puede observar que el factor de relajación propuesto no supera el resultado obtenido por SA en ninguna de las 30 ejecuciones, cuando el tamaño de maya es de 11x11.

En la Figura 4.5 se muestra de forma gráfica los resultados obtenidos en la Tabla 4.7, para los diferentes tamaños de malla con los factores propuestos y los obtenidos con SA, se compara el tiempo computacional con respecto al número de iteraciones que se requieren para resolver el problema hidrodinámico.

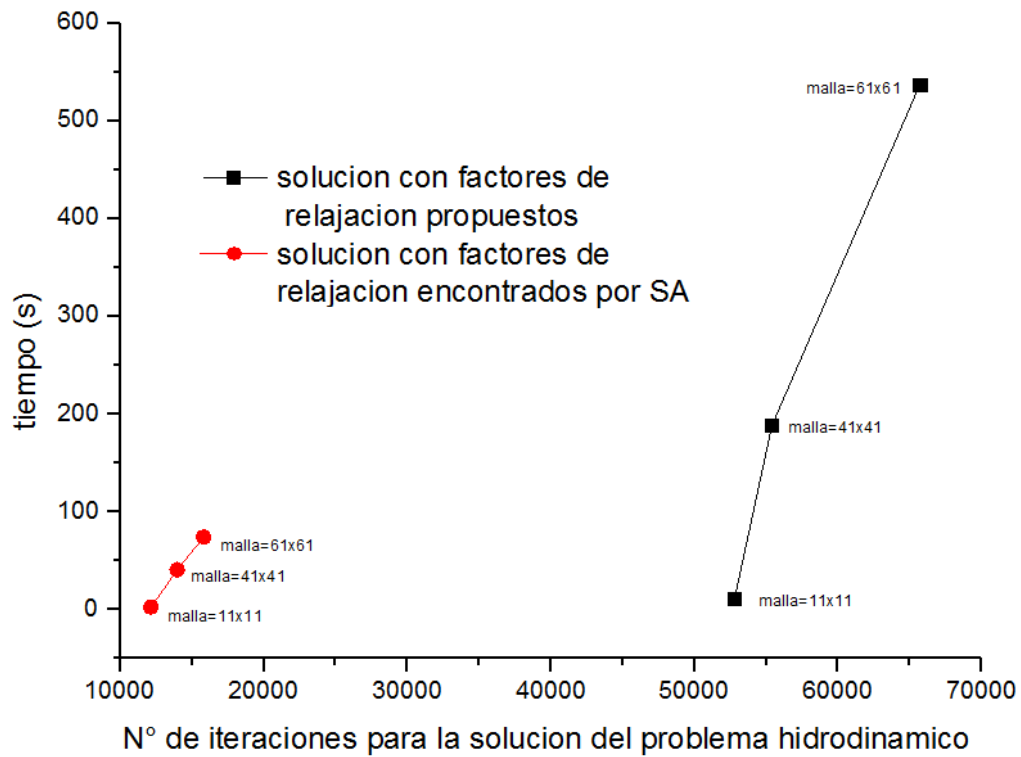


Figura 4.5. Comparación del tiempo computacional (t) con respecto al N° de iteraciones para la solución del problema hidrodinámico para diferentes factores de relajación (los propuestos y los encontrados con SA).

Se observa en la Figura 4.4 cualitativamente que al comparar los diferentes tamaños de malla con distintos factores de relajación varia el tiempo computacional que se requiere para dar solución al problema planteado, además, en cada uno de los casos cuando son utilizados los factores de relajación encontrados por SA se aprecia que se mejora el tiempo computacional y con ello se disminuye el número de iteraciones necesarias para darle solución al problema planteado para cada una de los diferentes tamaños de malla, caso contrario al utilizar los factores de relajación propuestos el tiempo computacional es mayor al igual que el número de iteraciones necesarias aun utilizando malla gruesa de 11x11.

Figura 4.6, muestra la convergencia del problema hidrodinámico (caso 2) de forma experimental con respecto al tamaño de la vecindad (LCM), en la implementación de recocido simulado al problema hidrodinámico.

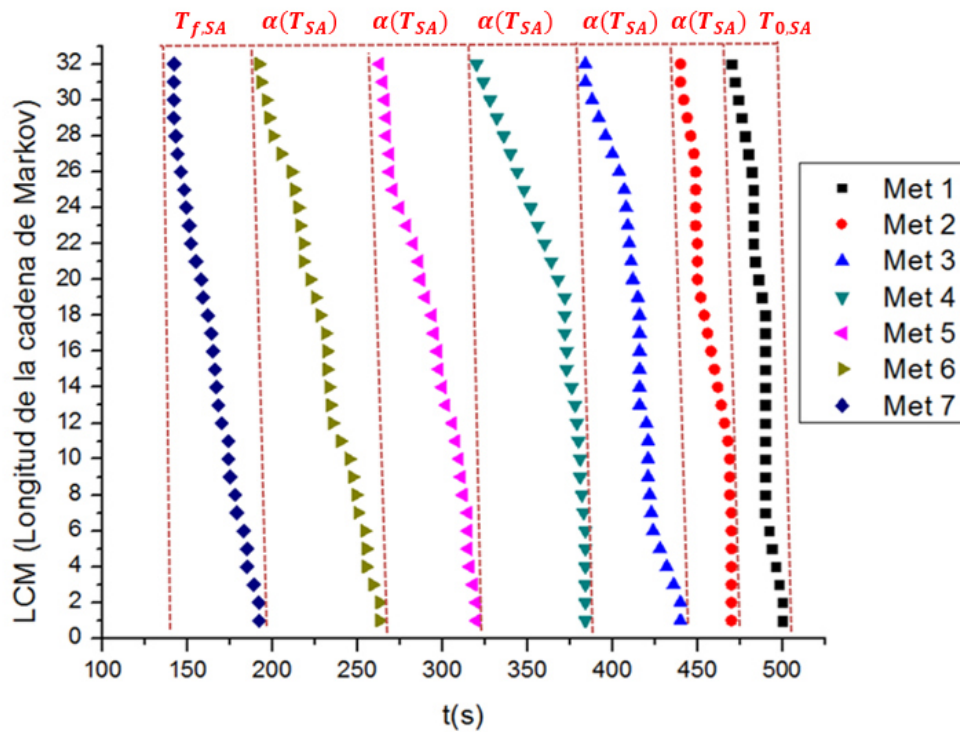


Figura 4.6. Tiempo de convergencia del problema hidrodinámico con SA

En la Figura 4.6 presenta la convergencia entre recocido simulado y el tiempo que se requiere para resolver el problema planteado, para cada Metrópolis identificado por la variable (Met) con respecto a un valor de temperatura constante se evalúan diferentes soluciones de acuerdo al espacio de vecindad $N(S_i)$, conforme decrece la temperatura de recocido simulado el tiempo de convergencia del problema hidrodinámico de calor se va reduciendo conforme RS va alcanzando su punto de enfriamiento, en el cual se logra el tiempo computacional más corto del problema hidrodinámico; es decir, para Met 1 se requieren más de 450 s para una solución, mientras que para Met 7 el tiempo

es menor a 150 s, con esto se comprueba que al aplicar la metodología de SA, esta reduce el tiempo de convergencia del problema hidrodinámico, si bien, son tiempos cortos los que se están evaluando cabe recordar que las pruebas se realizaron para valores relajados del problema (tamaño de malla y valor residual).

Una vez obtenidos los factores de relajación optimizados con valores relajados del problema se probaron cada una de las configuraciones obtenidas en las pruebas realizadas en el problema completo con la finalidad de comprobar, si aún, con el problema relajado la mejor configuración que se obtuvo con la heurística en comparación con factores propuestos daban mejores tiempos computacionales en el problema sin relajar las variables mencionadas, la configuración que obtiene menor tiempo computacional evaluada en el problema hidrodinámico se muestra en la Tabla 4.8.

Tabla 4.8. Factores de relajación obtenidos con SA, con malla de 61x61 y residual ($\varepsilon = 1 \times 10^{-10}$) no relajados, para el problema del caso 2.

$\varepsilon = 1 \times 10^{-10}$ y tamaño de malla = 61x61								% mejora en el tiempo
Factores de relajación propuestos			tiempo (s)	Factores de relajación encontrados con SA			tiempo (s)	
$Frelax_U$	$Frelax_V$	$Frelax_P$		$Frelax_U$	$Frelax_V$	$Frelax_P$		
0.2	0.2	0.1	528.543	0.82	0.87	0.27	90.08	82
0.2	0.2	0.1	528.543	0.43	0.84	0.51	103.635	80
0.2	0.2	0.1	528.543	0.87	0.58	0.42	96.535	81

Por los resultados obtenidos en la Tabla 4.8 se puede observar que con la implementación de Recocido Simulado, se obtienen nuevos factores de relajación los cuales al ser evaluados en el problema hidrodinámico ofrecen una mejora significativa en la convergencia pues se reduce hasta un 70% el

tiempo de cómputo con respecto a los factores que se proponen por simple deducción empírica o conocimiento previo acerca del tema.

Figura 4.7, muestra los resultados obtenidos para las componentes de velocidad con respecto al trabajo de Guia, 1982. Los resultados mostrados se obtienen con base en los factores de relajación optimizados por la metodología de SA. La línea continua muestra los resultados obtenidos por el presente trabajo y los puntos en color rojo muestra los obtenidos por la referencia citada, los cuales se observa de manera cualitativa que son similares en cuanto al comportamiento que sufren las velocidades dentro de la cavidad.

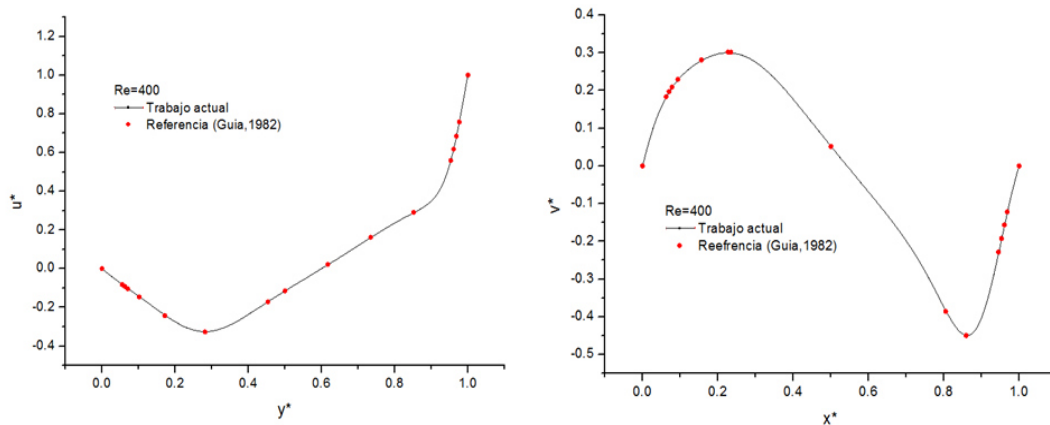


Figura 4.7. Componentes de velocidad dentro de la cavidad

4.2 Resultados experimentales algoritmo paralelo

Para las pruebas del algoritmo distribuido se utilizó un Clúster el cual se compone de un grupo de computadoras interconectadas y que cooperan entre sí para realizar una tarea en común, como se vio en el capítulo anterior uno de los objetivos de formar un clúster es el de obtener alto rendimiento en la resolución de un problema a través de la cooperación entre los equipos que lo

componen en un menor tiempo posible. Para el funcionamiento del clúster no es necesario que todos los nodos (cada uno de los procesadores) dispongan del mismo hardware y sistema operativo, debe disponer de una interfaz que se encargue de interactuar con el usuario y los procesos, repartiendo la carga entre los diferentes nodos del equipo.

4.2.1 Infraestructura del Clúster Cuexcomate

El clúster en el cual se realizó este proyecto fue el Clúster Cuexcomate el cual forma parte de la minigríd Tarántula; el clúster se puede ver en la Figura 4.8, está ubicado en el Laboratorio de Optimización y Software en el Centro de Investigaciones en Ingeniería y Ciencias Aplicadas (CIICAp) que pertenece al Instituto de Investigación en Ciencias Básicas y Aplicadas.



Figura 4.8 Clúster Cuexcomate.

En cual cuenta con el equipo y las siguientes características presentadas en la siguiente Tabla 4.9:

Tabla 4.9 Infraestructura Hardware y Software del Clúster Cuexcomate, Peralta 2016.

Nodos de procesamiento	
CPU 01 al 04	1 Motherboard <ul style="list-style-type: none"> • 2 procesadores Intel Xeon Six Core a 3.06 GHz, 12 MB cache • 1 HD Enterprise, 7200 RPM de 500GB
Total 48 cores	
Total 96 GB RAM	• 6 módulos RAM de 4GB 1333 MHZ DDR3. Total de 24GB RAM
Total 2 TB HD	1 tarjeta Infiniband 400Gb/s
Comunicaciones	
	Switch 3 COM 24/10/100/1000
	Switch Infiniband Mellanox de 18 puertos de 40 Gb/s
	1 Motherboard <ul style="list-style-type: none"> • 2 procesadores Intel Xeon Six Core a 3.06 GHz. 12 MB cache. • 2 HD Enterprise, 7200 RPM de 500GB (para S.O.). • 6 HD Enterprise, 7200 RPM, 12 TB en total. • 6 módulos RAM de 4 GB 1333 MHZ DDR3. Total de 24 GB RAM. • 1 tarjeta Infiniband 40 Gb/s.
Nodo de procesamiento	
GPU 05	1 Motherboard <ul style="list-style-type: none"> • 1 procesador Intel Xeon Six Core a 3.06 GHz, 12 MB cache. • 2 HD Enterprise, 7200 RPM de 500 GB.
Total 896 cores	<ul style="list-style-type: none"> • 9 módulos RAM de 4GB 1333MHZ DDR3. Total de 36 GB RAM. • 2 tarjetas NVIDIA TESLA C2070, arquitectura Fermi, con 6 GB RAM DDR5 c/u, 448 cores c/u.
Total 36 GB RAM	• DVD/RW
Total 1 TB HD	• Lector de memoria
	1 tarjeta Infiniband 40 Gb/s

Los recursos del clúster Cuexcomate se encuentran distribuidos como se indica en la Tabla 4.10.

Tabla 4.10 Distribución de los recursos del Clúster Cuexcomate.

Nodo	Cores
Cuexcomate	12
Ciicap01	12
Ciicap02	12
Ciicap03	12
Ciicap04	12
Ciicap-gpu01	6
Total	66

4.2.2 Resultados experimentales

Como ya se mencionó el proyecto se desarrolló en el cluster Cuexcomate, el nodo maestro es el Cuexcomate los nodos esclavos que fueron utilizados son: ciicap01, ciicap02, ciicap03 y ciicap04. Se realizaron 30 ejecuciones para el problema del caso 2, la distribución de los procesos fue de manera balanceada como se muestra en la Tabla 4.11, esto con la intención de no afectar la eficiencia del algoritmo esto puede ocurrir si algún nodo trabajara con menos procesos. La instrucción de trabajo para la distribución de carga fue la siguiente: `mpiexec.hydra -ppn <nodo> -n <número de procesos> -hostfile <archivo nombre nodos> ./<archivo a ejecutar>`

Tabla 4.11. Distribución de procesos de nodos Cuexcomate.

Total de procesos	# de procesos por nodo
4	1
8	2
16	4
32	8
64	16

La finalidad del cómputo paralelo es disminuir el tiempo de ejecución de un algoritmo, para ello existen diferentes métricas para este estudio se utilizó la

métrica más empleada que es el *Speedup* la cual permite comparar la ganancia de velocidad del algoritmo distribuido con respecto al algoritmo secuencial. El *Speedup* se calcula con la siguiente formula:

$$Speedup = \frac{\text{Tiempo de ejecución del algoritmo secuencial}}{\text{tiempo de ejecución del algoritmo paralelo}}$$

Tabla 4.12. Tiempos de ejecución del algoritmo distribuido del problema caso 2, para una malla de 11x11 y un $\varepsilon = 1x10^{-5}$.

# de procesos	Tiempo de ejecución secuencial (s)	Tiempo de ejecución paralelo (s)	Spedup	Speed-up Ideal
1	0.029			1
4		0.021	1.38	4
8		0.01	2.9	8
16		0.003	9.6	16
32		0.0015	19.3	32
64		0.001	29	64

En la Tabla 4.12 se observa que cuando se utiliza solo un proceso el tiempo de ejecución que se requiere para encontrar una solución al problema planteado para el caso 2 es de 0.029 segundos mientras que al implementar la distribución de procesos el tiempo disminuye hasta en más del 70% conforme se incrementa el número de procesos.

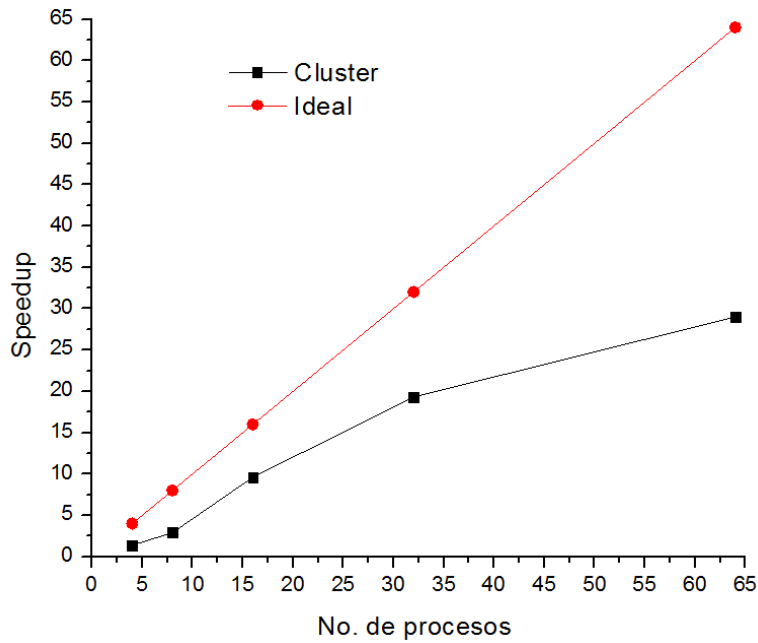


Figura 4.9. Comportamiento del Speedup

En la Figura 4.9 se muestra el comportamiento del *Speedup* en forma ideal con respecto *Speedup* cuando se hace la distribución de procesos, si bien el comportamiento que se obtiene no es lineal se observa que al incrementar el número de procesos se mejora el tiempo de ejecución del algoritmo.

CAPITULO V. CONCLUSIONES Y TRABAJOS FUTUROS

5.1 Conclusiones

Se concluye que la metodología propuesta de utilizar la meta heurística del algoritmo de SA, es una alternativa eficaz para encontrar los factores de relajación optimizados los cuales ayudan en la aceleración de convergencia los dos casos de prueba planteados, debido a que el comportamiento de cada una de las 30 ejecuciones de SA, presentan resultados de convergencia muy buenos.

Los resultados para el caso 1 muestran que se mejora hasta más de un 50% el tiempo de convergencia utilizando los factores de relajación encontrados por SA en comparación con los propuestos de forma aleatoria.

Para el caso 2, se observa que la mejora es más del 70% en tiempo computacional, utilizando la mejor configuración de los tres factores de relajación encontrados por el algoritmo heurístico SA en comparación con los propuestos de forma empírica, tomando en cuenta que, el proponer factores adecuados de forma empírica se debe tener cierto conocimiento del tema, sin embargo, con la heurística es posible encontrar factores adecuados sin necesidad de la experiencia.

Los tiempos de convergencia en ambos problemas (caso 1 y 2), no muestran un comportamiento directamente proporcional de forma clara y lo que si muestran es que cuando se realiza un pequeño incremento o de cremento en un factor de relajación, el tiempo de convergencia del problema puede aumentar o disminuir, para el caso 2 que trabaja con tres factores de relajación se hace más visible esta conclusión. Este comportamiento de la convergencia en función de los factores de relajación confirma lo que se presenta en la literatura.

5.2 Trabajos futuros

Con respecto a los resultados obtenidos en este proyecto de tesis doctoral, se plantean diversas actividades para realizar a futuro:

- Realizar un estudio más amplio de la metodología implementada en el área de mecánica de fluido computacional, probando con otros problemas de transferencia de calor, por ejemplo considerando convección natural convección forzada y convección mixta.
- Implementar nuevas estructuras de vecindad para la búsqueda de los factores de relajación y probar su efectividad.
- Buscar nuevas técnicas para la implementación de la heurística en la búsqueda de factores de relajación.
- Probar con otra (s) metaheurísticas en este tipo de problemas
- Comprobar la complejidad algorítmica de acuerdo a la clasificación de la teoría de la Complejidad en donde se encuentran ubicados este tipo de problemas.

REFERENCIAS

Aarts, E., & Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization*. Princeton: Princeton University Press.

A. Grama, G. K. (2003). *Introduction to Parallel Computing* (2da. Edition). Addison Wesley.

Aarts E., Korst J. (1989). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Gran Bretaña. pp. 272.

Azencott, R. (1992). *Simulated Annealing Parallelization Techniques*. Wiley and Sons.

Alonso-Pecina, F. (2008). *Programación de horarios escolares con Recocido Simulado y Búsqueda Tabú*. Cuernavaca, Morelos, México. ITESM.

Bonilla J., Yebra L., Dormido S., (2011). A heuristic method to minimise the chattering problem in dynamic mathematical two-phase flow models, *Math. Computer Modeling*, Vol. 54, págs. 1549-1560.

Brandt A., 1997, Multi-Level Adaptive Solutions to Boundary-Value Problem, *Math. Comput.*, Vol. 31, págs. 333-390.

Baotong Li, Jun Hong y Xiangyang Tian, “Generating optimal topologies for heat conduction by heat flow paths identification”, *International Communications in Heat and Mass Transfer* 75 (2016) 177–182.

Blaise Barney, L. L. (15 de 12 de 2014). *Message Passing Interface (MPI)*. Recuperado el 28 de 04 de 2015, de <https://computing.llnl.gov/tutorials/mpi/>.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, Vol. 35(3), 268-308.

Cortéz A. (2004). *Teoría de la Complejidad Computacional y Teoría de la Computabilidad*. *Revista Investigación y Sistemas de Información*. pp. 102-105.

Cruz-Chávez, M., Peralta-Abarca, J., & Moreno-Bernal, P. (marzo-julio de 2014). Aplicación de la teoría de la complejidad en Optimización Combinatoria. *Inventio*, 10(20), 35-42.

Díaz, B. A., & Dowsland, K. A., 2003. Diseño de Heurística y Fundamentos del Recocido Simulado. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7(19), 93-102.

Dragojlovic Z., Kaminski D. A., 2004, A Fuzzy Logic algorithm for acceleration of convergence in solving turbulent flow and heat transfer problems, Num. Heat Transfer., Part B.,Vol. 46, págs. 301-327.

Dréo, J., Pétrowski, P. S., & Taillard, E. (2006). *Metaheuristics for hard Optimization*. Berlin: Springer -Verlag.

Dréo , J., Pétrowski, P. S., & Taillard, E. (2006). *Metaheuristics for hard Optimization*. Berlin: Springer -Verlag.

Duarte, M., & Pantrigo, F. (2007). *Metaheurísticas*. Madrid, España.: Dykinson.

E-G. Talbi. Metaheuristics: from design to implementation. John Wiley and Sons, United Kingdom, 2009.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13:533–549, 1986.

García Regis, O. R., & Cruz Martínez, E. (2003). *Paralelización*. México, D.F.: UNAM.

Gallego, R., & Escobar, Z. (2008). *Técnicas Heurísticas de optimización* (2 ed.). (U. T. Pereira, Ed.) Pereira, Colombia: Textos Universitarios.

Ghia U. Ghia K. N., y Shin C. T., High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *Journal of Computational Physics*, vol. 48, pág. 387-411, 1982.

Gropp, W., Lusk, E., & Skjellum, A. (1995). Message Passing Interface Forum. *MPI: A Message Passing Interface Standard*.

G.L. Nemhauser, L.A. Wolsey. Integer and Combinatorial Optimization. John Wiley & Sons, New York, 1988.

Hidrobo, F. y. (2005). *Introducción a MPI (Message Passing Interface)*. Mérida, Venezuela: Centro Nacional de cálculo Científico, Universidad de los Andes.

H.K. Versteeg, W. Malalasekjera, "An Introduction to Computational Fluid Dynamics-The Finite Volume Method", Prentice, 2007.

Johnson, D.S, C.R. Aragon, L.A. McGeoch, S. Schevon (1989). Optimization by Simulated Annealing: En experimental evaluation, Part I, Graph Partitioning, Operations Research, 37(6) pp 865-892.

Juárez-Chávez, J., Cruz-Chávez, M., Serna-Barquera, S., Campillo-Illanes, B., Peralta-Abarca, J., Moreno-Bernal, P., & Martínez-Bahena, B. (2012). Neighborhood Hybrid Structure for the Optimization of Mechanical Properties of a Microalloyed Steel Based on its Chemical Composition. Electronics, Robotics and Automotive Mechanics Conference, CERMA2012, (págs. 20-23). Cuernavaca, Morelos.: IEEE-Computer Society.

J. Xamán, I. Zavala-Guillen, I. Hernández-López, J. Uriarte-Flores, I. Hernández-Pérez, E.V. Macías-Melo, K. M. Aguilar-Castro (2018). Evaluation of the CPU time for solving the radiative transfer equation with high-order resolution schemes applying the normalized weighting-factor method. Journal of Quantitative Spectroscopy & Radiative Transfer 208, pp 45-63.

J. Xamán, I. Hernández-López, J. Uriarte-Flores, I. Hernández-Pérez, I. Zavala-Guillen, P. Moreno-Bernal, J. F. Hinojosa (2018). X- factor: A modified relaxation factor to accelerate the convergence rate of the radiative transfer

equation with high-order resolution schemes using the Normalized Weighting-Factor method. *Computer Physics Communications* 231, pp 72-93.

Kirkpatrick, S. C.D. Gelatt Jr, M.P. Vecchi. (1983) Optimization by Simulated Annealing, *Science*, 220, pp 671-680

Malek, M., Guruswamy, M., Pandya, M., & Owens, H. (1989). Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Annals of Operations Research*, 21, 59-84.

Martínez-Bahena, B., Cruz-Chávez, M., Díaz-Parra, O., Martínez-Rangel, G., Cruz-Rosales, H., Peralta-Abarca, J., & Yanel-Juárez, J. (2012). Neighborhood Hybrid Structure for Minimum Spanning Tree Problem. *Electronics, Robotics and Automotive Mechanics Conference, CERMA2012* (págs. 191-196). Cuernavaca, Morelos.: IEEE-Computer Society.

Martínez-Oropeza. A. (2010). “Solución al Problema de Máquinas en Paralelo No Relacionadas Mediante un Algoritmo de Colonia de Hormigas”. Tesis de Maestría. Agosto 2010.

Martí, R. (2003). *Procedimientos Metaheurísticos en Optimización Combinatoria. Matemáticas*. Vol.1. No. 1. Pp. 3 – 62.

Martin Richard P., Vahdat Amin M., Culler David E., Anderson Thomas E. (1997). *Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture*. ISCA '97. Proceedings of the 24th Annual International Symposium on Computer Architecture. pp. 85-97.

Metropolis, N; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H. y Teller, E., 1953. Equation of state calculation by fast computing machines. *Journal of Chemistry Physics*, 21:1087-1091.

Márquez Gómez, Mervyn (2014). *Las metaheurísticas: tendencias actuales y su aplicabilidad en la ergonomía*. Ingeniería Industrial. Actualidad y Nuevas

Tendencias, Vol. IV, N° 12. ISSN: 1856-8327. Disponible en: <https://www.redalyc.org/articulo.oa?id=2150/215037911009>.

Moreno Díaz, P., Huecas Fernández-Toribio, G., Sánchez Allende, J., & García Manso, A. (2007). Metaheurísticas de optimización combinatoria: Uso de simulated annealing para un problema de calendarización. *Tecnología@ y Desarrollo. Revista de ciencia, Tecnología y Medio Ambiente*, 1-25.

Nourani, Y., & Andresen, B. (1998). A comparasion of simulated annealing cooling strategies. *J. Phys. A: Math. Gen.*(31), 8373-8385.

Osborne L.J., B.E. Gillet., (1991). A comparison of two simulated annealing algorithms applied to the directed Steiner problem on networks. *ORSA Journal on computing* 3(3) 2013.

Osman, I. (1993). Metastrategy Simulated Annealing and Tabu Search Algoritms for the Vehicle Routing Problem. *Annals of Operations Research*, 41, 421-451.

Pacheco Peter S. (1997). *Parallel Programming with MPI*. Morgan Kaufmann Publishers. ISBN. 1-55860-339-5. pp. 293.

Papadimitriou C. H., Steiglitz K. (1998). "Combinatorial Optimization: Algorithms and Complexity". ISBN. 0-486-40258-4. USA. 1998. Mineola NY.

Patankar S. (1980), "Numerical Heat Transfer and Fluid Flow", Hemisphere Publishing Co., Mc. Graw Hill Co., New York.

Patankar S., Spalding D., 1972. A calculate procedure for heat mass y momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, Vol. 15, pages 1787-1806.

Papadimitriou C. H., Steiglitz K. (1998). "Combinatorial Optimization: Algorithms and Complexity". ISBN. 0-486-40258-4. USA. 1998. Mineola NY.

Peralta-Abarca, J. Enríquez-Urbano, Martínez-Bahena, B., Alfonso D'Granda Trejo (marzo-junio de 2018). Esquema de enfriamiento en la metaheurística de recocido simulado. *Inventio*, 10(20), 37-42.

Peralta-Abarca. (2016). "Algoritmo distribuido de Recocido Simulado para el modelo del transporte vehicular con capacidades homogéneas". Tesis de Maestría. Doctorado.

Ram, D., Sreenivas, T., & Subramaniam, K. (1996). Parallel Simulated Annealing Algorithms. *Journal of Parallel and Distributed Computing* (37), 207-212.

Reeves, C. R. (1996). Modern Heuristics Techniques. (I. O. Rayward-Smith V.J., Ed.) *Modern Heuristics Search Methods*, 1-25.

Ryoo J., Kaminski D., Dragojlovic Z., 1999, Automatic Convergence in a Computational Fluid Dynamics Algorithm Using Fuzzy Logic, The Sixth Annual Conference of the Computational Fluid Dynamics Society of Canada, VIII.

Saad Y., Schultz M. R, 1986, GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comp.*, Vol. 7, págs. 856-869.

Sudhakar T., Balaji C., Venkateshan S.P., 2010, A heuristic approach to optimal arrangement of multiple heat source under conjugate natural convection, *Int. J Heat Mass Transfer*, Vol. 53, págs. 431-444.

S. Kirkpatrick, C.D. Gelatt Jr, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598), 13 May, págs. 671-680, 1983.

Ullah A, Malik SA, Alimgeer KS (2018) Evolutionary algorithm based heuristic scheme for nonlinear heat transfer equations. *PLoS ONE* 13(1): e0191103. <https://doi.org/10.1371/journal.pone.0191103>

Van Laarhoven, P., Aarts, E., & Korst, J. (1997). Simulated Annealing. En E. Aarts, & J. Lenstra, *Local search in combinatorial Optimization* (págs. 91-120). John Wiley & Sons Ltd.

Versteeg H. K y Malalasekera, W. An introduction to computational fluid Dynamics. The finite volume method (1a ed.; United States, New York, Longman Scientific & Technical 1995).

Yainier Labrada-Nueva, J. Enríquez-Urbano y Yadián García-Ojito, (2014). “Un algoritmo de búsqueda local iterada como solución al problema de la mochila”. Programación Matemática y Software. Págs. 57-64.