



Article Metaheuristic to Optimize Computational Convergence in Convection-Diffusion and Driven-Cavity Problems

Juana Enríquez-Urbano ¹, Marco Antonio Cruz-Chávez ^{1,*}^(D), Rafael Rivera-López ²^(D), Martín H. Cruz-Rosales ³^(D), Yainier Labrada-Nueva ¹ and Marta Lilia Eraña-Díaz ¹^(D)

- Research Center in Engineering and Applied Sciences, Autonomous University of Morelos State (UAEM), Cuernavaca 62209, Mexico; juana.enriquez@uaem.mx (J.E.-U.); yainier.labrada@uaem.mx (Y.L.-N.); merana@uaem.mx (M.L.E.-D.)
- ² Computation and Systems Department, National Technological Institute of Mexico/Veracruz Technological Institute, Veracruz 91860, Mexico; rrivera@itver.edu.mx
- ³ Faculty of Accounting, Administration & Informatics, UAEM, Cuernavaca 62209, Mexico; mcr@uaem.mx
- Correspondence: mcruz@uaem.mx

Abstract: This work presents an optimization proposal to better the computational convergence time in convection-diffusion and driven-cavity problems by applying a simulated annealing (SA) meta-heuristic, obtaining optimal values in relaxation factors (*RF*) that optimize the problem convergence during its numerical execution. These relaxation factors are tested in numerical models to accelerate their computational convergence in a shorter time. The experimental results show that the relaxation factors obtained by the SA algorithm improve the computational time of the problem convergence regardless of user experience in the initial low-quality *RF* proposal.

Keywords: overlaps; neighborhood structure; amorphous shapes; paper waste; resource allocation; perturbations

1. Introduction

In the present day, Computational Fluid Dynamics (CFD) is a powerful tool for analyzing and understanding various physical phenomena occurring in nature and in industrial processes, which are objects of study in various fields of research. Generally, the numerical solution of fluid flow, heat transfer and mass problems require an iterative process, mainly due to their nonlinearity. In particular, for the development of computational simulations in mechanical engineering, it is necessary to solve a set of partial differential equations (PDE) using numerical techniques such as finite-difference, finite-element and finite-volume methods. These methods convert a PDE to an algebraic equation system in a discrete domain and find an approximate solution to the original problem. For example, to simulate fluid-flow phenomena, or heat and mass transfer issues, they are first modeled as nonlinear problems and then solved using iterative processes [1]. These processes commonly require high computational resources using robust convergence criteria, allowing them to find a solution that satisfies the problem conditions. One of the most commonly used criteria is to verify that the difference between the current and the previous solution is minimal. Furthermore, the rate-of-change of the dependent variable (velocity, pressure or temperature, for example) in the iterative process can be modified using over-relaxation if this rate is accelerated or under-relaxation if it is reduced. In particular, under-relaxation is very useful to solve nonlinear problems since it avoids iterative process divergence.

The study of the convergence properties of discretization-based methods to solve nonlinear problems is essential to design effective procedures to reach solution in acceptable times. The design of efficient techniques and algorithms solving complex optimization problems has become one of the most important engineering research fields. Moreover, the computational costs (spatial and temporal) are critical factors that must be addressed



Citation: Enríquez-Urbano, J.; Cruz-Chávez, M.A.; Rivera-López, R.; Cruz-Rosales, M.H.; Labrada-Nueva, Y.; Eraña-Díaz, M.L. Metaheuristic to Optimize Computational Convergence in Convection-Diffusion and Driven-Cavity Problems. *Mathematics* 2021, *9*, 748. https:// doi.org/10.3390/math9070748

Academic Editor: Raimondas Ciegis

Received: 8 February 2021 Accepted: 22 March 2021 Published: 31 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). when solving large size and complex problems; therefore, several related studies have been published in the scientific literature. For example, in reference [2], the temperature distribution in a guarded hot plate apparatus used to measure the thermal conductivity of insulating materials was analyzed. For validating the experimental results, authors compared both analytical and numerical solutions and reported that the time required to reach a steady-state temperature distribution was nearly 640 min. In reference [3], a numerical solution of heat transmission problems with phase change is proposed. The authors compare two algorithms to solve engineering problems: one is analytical, and the other discretizes the equations governing the problem. By using relaxation factors empirically, they first apply subrelaxation to stabilize the algorithm convergence and then use over-relaxation to accelerate it. To obtain a sufficient temperature history for phenomenon interpretation, and approximate time of 48 days is necessary.

Several methods trying to reduce the convergence time to solve heat transfer problems have been proposed in light of these facts. One of them is the multigrid acceleration technique [4], which improves the convergence rate when solving CFD simulations with large algebraic equation systems based on a numerical mesh modification. Furthermore, in references [5,6], the Navier-Stokes equations modeling a two-dimensional Driven-Cavity problem are solved using the SIMPLE and SIMPLE Revised methods, and the Vorticity-Stream function approach. The computational results indicate that each method's convergence acceleration depends on the Reynolds numbers and the mesh size.

In reference [7], a genetic algorithm (GA) is applied to solve the water-steam flow thermohydraulic model of a direct solar steam system. The authors implement a chattering detector to minimize the phase changes and avoid the process divergence. This approach reduces the computational time up to 85% with a maximum error of less than 2%. In references [8,9], the convergence rate to solve a radiative transfer equation with several high-order and high-resolution schemes, applying the normalized weighting-factor method (WFM), is evaluated. In reference [8], the WFM convergence time is compared with that obtained by a deferred correction (DC) technique. Experimental results indicate a time reduction rate of 23.1%, 12.6% and 56.1% of WFM over DC, using three different schemes. In reference [9], the X-factor method is compared, in terms of computer time needed to obtain a converged solution, with the widely used deferred-correction (DC) method for calculating a two-dimensional cavity with emitting–absorbing–scattering gray media using the discrete ordinates method. A time reduction rate of up to 211%, 181% and 219% is obtained. This solution is relative to the solution time obtained compared to the deferred-correction method.

In references [10,11], optimal relaxation factors in the numerical solution of CFD problems are proposed. The convergence acceleration is achieved by automatically adjusting the relaxation factors during the code execution based on the behavior of the residuals, using a fuzzy logic technique, orienting under-relaxation in the discretized Navier-Stokes equations. This method is evaluated by solving natural convection problems in square cavities. The computational efficiency of this method is verified by comparing it with the use of constant relaxation factors. A 35% reduction in convergence time is obtained for the natural convection phenomenon. In reference [12], three CFD models of basic cases of natural, forced and mixed convection in closed environments are depicted. A procedure modifying parameters such as time step, mesh size and relaxation factors are also evaluated. The authors analyze diverse variables such as pressure, density, body forces, momentum, turbulent kinetic energy, specific dissipation rate, turbulent viscosity and energy. Experimental analysis indicates that it is necessary to establish correct time steps and relaxation factors to reduce the computational time in the convergence process, which implies advanced knowledge of the problem and modeling expertise. In general, when relaxation factors are proposed to ensure convergence, experience of those who are solving the problem is required, and sometimes this experience improves computational time but not the optimal solution. The relaxation factors must be between zero and one. A factor value nearest to zero ensures convergence but high computational time. On the

other hand, an amount closest to one accelerates the solution convergence, but it may tend to diverge. In these cases, the use of heuristics and metaheuristics are viable alternative approaches to solve these problems since they use computational resources efficiently.

To the best of our knowledge, no proposals have been found that optimize the convergence time for convection-diffusion and driven-cavity problems with computational heuristics. The closest study found in the existing literature was that using fuzzy logic. It is clear that with the use of heuristics it is no longer necessary to have experience in proposing good relaxation factors empirically.

The works presented in this introduction are the only works that are close to the research presented in this one.

In this paper, a simulated annealing-based approach to optimize relaxation factors and improve computational convergence in solving Convection-Diffusion and Driven-Cavity problems is described. These relaxation factors are used as input data to apply under-relaxation in the numerical methods solving Convection-Diffusion and Driven-Cavity problems. Experimental results show that the computational convergence time is improved by up to 70% compared with tests carried out with the relaxation factors applied without first optimizing them with simulated annealing. The present work only focuses on the study to increase the convergence of Convection-Diffusion and Driven-Cavity problems. This work aims to present a heuristic goal to improve the convergence time in convection-diffusion and driven-cavity problems without having the experience of empirically proposing good relaxation factors. The main contribution is the adaptation of the heuristic goal of the simulated annealing (SA) algorithm to find relaxation factors in convection-diffusion and driven-cavity problems, so as to improve the convergence of these problems. No similar method to the one proposed in this work has been found in the literature.

The rest of this paper is organized as follows: Section 2 describes the physical and mathematical models of the two problems addressed in this work. Section 3 presents the proposed methodology to find the optimized relaxation factors helping solve these problems with a reduction in the computational convergence time. Section 4 shows the experimental results in which it is verified that the use of a simulated-annealing-based approach obtains relaxation factors reducing the computational convergence time in the problem convergence. Finally, Section 5 shows the conclusions derived from this work.

2. Paper Case Studies Definition

Two fluid-dynamics problems are evaluated: The Convection-Diffusion and the Driver-Cavity problems. In the first one, the temperature is the dependent variable, and since the velocities are considered constant, only one relaxation factor is utilized. In the second one, the variables are the velocities and the pressure, so in this case, three relaxation factors are needed.

2.1. The Convection-Diffusion Problem

This section shows an example to determine the temperature variation in a transient state, T(x, y, t), in a two-dimensional domain, with constant properties: $\rho = 2702 \text{ kg/m}^3$, $C_p = 903 \text{ J/kg}^\circ\text{C}$, and $\lambda = 237 \text{ W/m}^\circ\text{C}$. It is considered that (1) the variable is transported by heat convection and conduction, and that through the dimensional medium, there is no heat generation, and (2) the velocity components (u,v) are constant at any point in the physical domain with: u = 2 m/s and v = 2 m/s. The medium is geometrically square with a length of one meter H = 1 m, subject to boundary conditions, as shown in Figure 1. For the diffusion convection problem, only the heat equation is solved.



Figure 1. Heat convection-conduction problem (physical model) [13].

The mathematical model for case 1 presents the general convection-diffusion equation for the variable of interest \emptyset in the Cartesian system (Equation (1)). In this case, the variable of interest is the temperature (*T*) (Equation (2)). By using the finite volume technique, the discretization of the proposed mathematical problem is carried out, which is presented in (Equation (3)) as the discrete equation in grouped coefficient notation [13].

$$\frac{\partial(\rho\otimes)}{\partial t} + \frac{\partial(\rho u\otimes)}{\partial x} + \frac{\partial(\rho w\otimes)}{\partial y} = \frac{\partial}{\partial x} \left(\Gamma \frac{\partial\otimes}{\partial x}\right) + \frac{\partial}{\partial y} \left(\Gamma \frac{\partial\otimes}{\partial y}\right) + S \tag{1}$$

$$\frac{\partial(\rho T)}{\partial t} + \frac{\partial(\rho u T)}{\partial x} + \frac{\partial(\rho v T)}{\partial y} = \frac{\partial}{\partial x} \left(\frac{\lambda}{C_p} \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y} \left(\frac{\lambda}{C_p} \frac{\partial T}{\partial y}\right) + S$$
(2)

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b \tag{3}$$

where S = 0, $a_E = D_e A(|Pe_e|) + max[-F_e, 0]$, $a_W = D_w A(|Pe_w|) + max[F_w, 0]$, $a_N = D_n A(|Pe_n|) + max[-F_n, 0]$, $a_S = D_s A(|Pe_s|) + max[F_s, 0]$, $a_P = a_E + a_W + a_N + a_S + a_P^0$, $a_P^0 = \rho_P^0 \frac{\Delta x * \Delta y}{\Delta t}$, and $b = a_P^0 T_P^0$. The function $A(|Pe_e|)$ is a function depending on the numerical scheme that is being used, in this case power law, where D_e , D_w , D_n , D_s are diffusive flows in the interface of the control volume; F_e , F_w , F_n , F_s are convective flows; a_E , a_W , a_N , a_S are the coefficients of the discretized equation and b is the source term. The power law scheme uses an exponential approximation, which improves the accuracy of the method at a lower computational cost [13].

2.2. The Driven-Cavity Problem

The physical model of the hydrodynamic problem with its boundary conditions is shown in Figure 2. It is a laminar and incompressible flow regime in a square cavity. The fluid has $\mu = 1.817 \times 10.5$ kg/(m s), and $\rho = 1.2047$ kg/m³. The cavity size is $H_x = H_y = 1$ m, and its upper wall moves with a uniform velocity $U_0 = 1.508 \times 10^{-3}$ m/s, corresponding to a Reynolds number of 400. The Driven-Cavity Problem is commonly used as a test case because of its nonlinearity property. It can present instability when looking for a numerical solution, and to reach convergence requires extensive modeling experience.



Figure 2. Hydrodynamic problem, Driven-Cavity (physical model) [13].

In fluid dynamics problems, velocity components and a pressure gradient are involved. The equations governing the hydrodynamic phenomenon are the mass and momentum conservation equations (Equations (4)–(6) with Fx = Fy = 0) to couple the pressure variable's determination to resolve the hydrodynamic equations. In [13,14], they proposed the SIMPLE algorithm to solve the two-dimensional case for a Newtonian and incompressible fluid in a laminar regime. The resulting equations, adapted from the general convection-diffusion equation (Equation (1)), are the mass equation (Equation (4)), the Momentum equation u (Equation (5)), and the Momentum equation v (Equation (6)), where ρ is the density of the medium, u and v are the speed components, t is the time and μ is the dynamic viscosity. By using the finite volume technique, the discretization of the proposed mathematical model is performed, and the result is represented in Equation (7) as the discrete equation in grouped coefficient notation [13,14], where $a_E = \rho_e d_e^u \Delta y$, $a_W = \rho_w d_w^u \Delta y$, $a_N = \rho_n d_n^v \Delta x$, $a_S = \rho_s d_s^v \Delta x$, $a_P = a_E + a_W + a_N + a_S + a_p^0$ and $b' = \frac{(\rho - \rho_P^0)}{\Delta t} \Delta x \Delta y + [(\rho_w u_w) - (\rho_e u_e)] \Delta y + [(\rho_s v_s) - (\rho_n v_n)] \Delta x$. The coefficients $\left(d_e^u = \frac{A_e}{a_e^u} y d_w^u = \frac{A_w}{a_w^u}\right)$, and $\left(d_n^v = \frac{A_n}{a_n^v} y d_s^v = \frac{A_s}{a_s^v}\right)$ represent the relationship between correction speeds and correction pressure, respectively.

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0$$
(4)

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u u)}{\partial x} + \frac{\partial(\rho v u)}{\partial y} = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) - \frac{\partial p}{\partial x}$$
(5)

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho u v)}{\partial x} + \frac{\partial(\rho v v)}{\partial y} = \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y}\right) - \frac{\partial p}{\partial y} \tag{6}$$

Using the finite volume technique, the discretization of the proposed mathematical model is as follows:

$$a_P P'_P = a_E P'_E + a_W P'_W + a_N P'_N + a_S P'_S + b'$$
⁽⁷⁾

To solve the Driven-Cavity hydrodynamic problem, in this work the SIMPLE algorithm presented in [14] is used.

3. Methodology to Find Relaxation Factors with Simulated Annealing

An optimization problem is one whose goal is to find a solution $s \in S$ with a minimum cost value $f(s) \in R$, where S is a finite set of candidate solutions and f is the objective function $f:S \to R$, which assigns a value representing the quality of each s. In particular, s * is the optimum global, if $f(s *) \leq f(s)$, $\forall s \in S$. For this work, each s is built utilizing the proposed values for the relaxation factors, and f(s) represents the convergence time obtained for s. Then, to determine more appropriate relaxation factor values to reach minimal convergence time in solving fluid-dynamics problems, a simulated annealing (SA) algorithm is used, since it is an effective metaheuristic to find a near-optimal solution to complex optimization problems in a reasonable time.

3.1. Simulated Annealing to Accelerate Convergence Time

The SA algorithm is one of the most widely used metaheuristics to solve complex optimization problems [15,16]. It is inspired by the metal annealing process used in industry, where the material properties are altered, until they reach an optimal state, by controlling the temperature [17,18]. The annealing process begins with heating the material to a high temperature and then cooling it slowly, maintaining a constant temperature at each stage for a specific time, thereby achieving the material's best physical configuration. The SA algorithm uses the procedure proposed by Metropolis [19] to simulate the cooling process and carry out local searches and, to escape local optimums, allows some worse movements. Table 1 shows the analogy between metallurgy concepts, combinatorial optimization and the heat transfer problem raised in this work.

Metallurgy	Combinatorial Optimization	Heat Transfer Problem
Configuration	Feasible solution	The heat transfer problem solution complies with the constraint satisfaction model
Energy configuration	Solution cost	Convergence time in the solution of the heat transfer problem
Minimum energy	Minimum value obtained with the objective function	Minimum convergence time (<i>t_{conv}</i>) based on the discrete equation of grouped coefficients and proposed relaxation factors. Equation (6)
Fundamental configuration	Optimal solution	Relaxation factors that obtain the minimum value of t_{conv}
Temperature	Control parameter T_{SA}	Control parameter T_{SA}
Thermodynamic equilibrium	Markov chain length MCL, in each Metropolis cycle (Met <i>i</i>), with <i>i</i> = 1 to MCL	Neighborhood size defining the number of neighboring solutions.
Temperature decrement	Control coefficient α	Control coefficient decrementing T_{SA}
Final temperature	Stopping criterion	Stopping criterion, the minimum value reaching T_{SA}
Metastable state	Optimal local	Local optimal solution obtained using relaxation factors to optimize t_{conv} .
Steady state	Global optimal solution	Global optimal solution obtained using relaxation factors to optimize t_{conv} .

Table 1. Analogy between the simulated annealing process in metallurgy and simulated annealing in combinatorial optimization.

3.2. Cost Function to Optimize Relaxation Factors

If relaxation factors are incorporated in the discretized Equations (3) and (7), the following equation is obtained:

$$s = a_P^{RF} \varnothing_P - [a_E \varnothing_E + a_W \varnothing_W + a_N \varnothing_N + a_S \varnothing_S] + b^{RF}$$
(8)

The goal is to minimize the convergence time of Equation (8). The adequate configuration of relaxation factors values, by increasing or decreasing them, generates the solution in a reduced time. For each iteration of the Metropolis cycle (Met *i*, with *i* = 1 to MCL) in the SA algorithm, a candidate solution *s* is obtained using equation (8) with the relaxation factors (*RF*) from a neighborhood structure described below. When solving the problem, there is a convergence time t_{conv} , defined as follows:

$$t_{conv} = f(s, RF) \tag{9}$$

The objective function minimizing the convergence time is as follows:

$$\min t_{conv} = \min f(s, RF) \tag{10}$$

3.3. Strategy to Optimize Relaxation Factors

The SA algorithm helps find the best configuration for different relaxation factors minimizing the problem of convergence times. In the first problem, a relaxation factor is sought for temperature, while in another three relaxation factors are needed, one for pressure and the remaining for each velocity component, u and v. By having the best relaxation factor values, it is possible to accelerate the convergence time to solve the convection-diffusion and driven-cavity problems.

The methodology steps for reducing the convergence time utilizing the SA algorithm are the follows:

- 1. Use Equation (10) as a cost function to conduct the SA iterative process.
- 2. Apply small perturbations in the relaxation factors for obtaining neighbor solutions of s.
- 3. Mesh size relaxation: coarse mesh sizes are tested for partial results. For example, if the appropriate mesh size for this problem obtained by a mesh independence analysis is 61×61 , its value is relaxed with coarse mesh values of 11×11 and 41×41 .
- 4. Residual value relaxation: larger residual values are evaluated for partial results. For example, if the appropriate residual value to achieve problem convergence is $\varepsilon = 1 \times 10^{-10}$, its value is relaxed to a larger value of $\varepsilon = 1 \times 10^{-3}$.
- 5. At the end of the SA execution, the optimized relaxation factors are obtained. These factors are compared with the appropriate values to evaluate the reduction of the computational convergence time.

3.4. Neighborhood Structure

The theory of the SA algorithm shows that, for it to work correctly, any solution must be reached from any other through a series of valid disturbances or movements [20,21]. The neighbourhood of a candidate solution s, named N(s), is the set of possible new solutions generated by applying small perturbation in s. For this work, the perturbations are applied to the relaxation factors to reduce convergence time and to reach a problem solution. An important decision in the SA algorithm is the neighborhood structure definition, i.e., the way to establish a new solution (a type of disturbance). The SA theoretical results show that it is enough to demand that any solution can be reached from another solution through a series of valid movements called perturbations, or movements [21,22]. For this work, the neighborhood structure does not perform disturbances since it chooses a decreased or increased value of ± 0.02 (value obtained through a sensitivity analysis of this parameter based on the value of the valid interval of 0 < RF < 1.). The 0.02 tuned value allows a fast SA convergence without being trapped in a low-quality optimal local solution. For example, if the value is decreased from 0.02 to 0.002, the SA is slower to find an optimal time of convergence, according to the *RF* found with SA for the convection-diffusion and driven-cavity problems analyzed in this work. If more than one relaxation factor is disturbed, one of them is first selected at random, and then it is updated, leaving the rest unchanged.

3.5. Simulated Annealing Algorithm

Algorithm 1 presents the implementation of the SA-based approach to finding a near-optimal configuration of the relaxation factors to solve convection-diffusion and driven-cavity problems. At first, the algorithm diversifies the search by accepting better or worse solutions interchangeably, but, as the iterative process progresses, it is increasingly challenging to accept worse solutions, due to the SA acceptance criterion.

Algorithm 1. The SA-based algorithm to solve convection-diffusion and driven-cavity problems.
1. Select an initial control parameter $T_0 > 0$
2. $T_{SA} = T_0$
3. α = Cooling velocity
4. MCL = Neighborhood size
5. T_f = Stop criterion value
6. $s = Initial \ solution \ (RF)$
$f(s,RF) = Solution \ cost$
N(s,RF) = Neighborhood function
7. repeat // External cycle
8. repeat // Internal cycle
9. select $s' \in N(s, RF)$
s' = Solution with new relaxation factors
10. $\delta = f(s',RF) - f(s,RF)$
11. if $\delta < 0$ then $s = s'$
12. else
13. Generate randomly $u \in U$
14. if $u < e^{-\delta/T_{SA}}$ then $s = s'$
15. end if
16. until reaching MCL
$17. T_{SA} = \alpha (T_{SA})$
18. until $T_{SA} \leq T_f$

First, the initialization steps are carried out in lines 1–6. The value of the control parameter T_0 must ensure a high probability of accepting worse movements initially. This value is obtained by identifying the best value positively influencing the solution quality [23]. T_{SA} is the variable controlling the SA external cycle, α is the coefficient used to adjust T_{SA} in each iteration, T_f represents the stop criterion value and s is the initial candidate solution. Furthermore, the Markov Chain Length (MCL) is used to control the SA internal cycle, allowing a stable state to be reached for each T_{SA} value. The neighborhood size is MCL = $NF \times CN$, where NF is the number of relaxation factors and CN is the number of factor changes. In this work, two changes are used: increasing and decreasing a relaxation factor.

The SA external cycle is implemented in lines 7–18. It includes the internal cycle and the decrease of T_{SA} until it reaches the stop value. The SA internal cycle is shown in lines 8–16, which is controlled by the MCL value. First, a neighbor solution s' is computed according to the neighborhood structure, its cost value f(s',RF) is calculated, and the difference between the current solution cost value is obtained. Lines 11–15 define the acceptance criterion for a new solution. If s' improves the solution cost, it replaces s. Otherwise, an s' is accepted as the current solution using Boltzmann's probability function. This acceptance criterion allows for the escape from one optimum local, which makes SA more efficient. It is important to emphasize that with acceptance probability having high

 T_{SA} values, most of the neighboring solutions are accepted, and for low T_{SA} values, worse movements are rejected.

4. Computational Results

This computational study was carried out on a computer with Intel Core i7-3612QM 2.10 GHz CPUs and 8 GB RAM, using a 64-bits Windows 7 SO, and Visual C++ 2010.

The results could not be compared with previous works because no techniques were found proposing to improve convergence for convection-diffusion and driven-cavity problems. The closest approach found in the existing literature to improve these problems' convergence were techniques using Reynolds numbers and mesh size. In our case, the convergence acceleration depends only on finding the optimal *RF*.

4.1. CASE 1: Convection-Diffusion Problem

Figure 3 shows the mesh independence analysis for the problem convergence, where it is observed that after the 101×101 mesh, qualitative differences are minimal. This case can be verified qualitatively in [13], where is shown the exact solution of the problem.



Figure 3. Temperature profile for CASE 1: Convection-diffusion problem.

Table 2 shows the convection-diffusion problem analysis where the *RF*s that help to converge the solution more efficiently are varied. It is observed that the best results are obtained when the value of both parameters is the highest (*RF* and Δt). Convergence analysis is observed in time when the parameters that help to converge the solution more efficiently to an adequate Δt of time are changed.

Table 3 shows the tuned values of the SA control parameters for the Convection-Diffusion problem.

<i>RF</i> (Relaxation Factor)	Δt	Time (s)	Iterations	Maximum Error
0.5	0.1	2.615	377	$9.1 imes 10^{-11}$
0.9	0.1	0.685	80	$9.6 imes10^{-11}$
0.4	1	3.22	526	$9.9 imes10^{-11}$
0.5	1	2.338	363	$8.73 imes10^{-11}$
0.9	1	0.597	71	$7.6 imes10^{-11}$
0.4	10	3.181	523	$9.9 imes10^{-11}$
0.5	10	2.304	362	$8.2 imes10^{-11}$
0.9	10	0.526	70	$7.8 imes10^{-11}$

Table 2. *RF* and Δt variation and errors obtained with a 101 × 101 mesh.

Table 3. Simulated annealing (SA) parameter values for the convection-diffusion problem.

Parameter	Value
T_0	2.00
T_{f}	0.01
MČL	2.00
α	0.965

Figure 4 shows the sensitivity analysis for Table 3 values. The acceptance probability concerning the number of SA iterations for the convection-diffusion problem, and one uniform distribution of the worse solutions through the algorithm execution at equal iteration intervals, are shown. It is observed that in the 90–100 probability interval, the number of worse solutions is small, unlike for the 0–10 range, where the number of these solutions is high. In Figure 4, each red point represents a solution generated by the Convection-diffusion problem in each SA iteration. The time it takes for SA to run is approximately 14 min, applying a mesh and relaxed to the Convection-diffusion problem. The convergence time of the convection-diffusion problem (Equation (10)) is evaluated as the SA cost function.



Figure 4. Sensibility analysis for SA parameters.

These values indicate that the number of worse solutions accepted during the entire execution of SA is minimal compared to the number of better solution accepted. When SA begins to iterate the probability of accepting worse solutions, it decreases until reaching the convergence in iteration 105 where the acceptance rate of worse solutions is not more than 10%. Then, at the end of the SA execution, this process stabilizes the search of solutions improving the current solution. It is essential to accept some worse solutions at the beginning of the SA algorithm so that the search process escapes from optimum locals and allows finding better solutions at the end of its execution. With SA's behavior presented in Figure 3, the quality of the tuned parameters shown in Table 3 is verified.

Since SA is a stochastic method, it is necessary to analyze results with a minimum of 30 executions for each problem, and for each of the relaxed factors. Table 4 shows the convection-diffusion problem results, with a 41 × 41 mesh and a residual value of $\varepsilon = 1 \times 10^{-5}$. The optimized relaxation factor for the temperature (RF_T) and the convergence time (t_{conv}) are also displayed. The range obtained is for 30 executions of t_{conv} is 0.025 s, with 0.01, 0.026 and 0.019 s as minimum, maximum and average times, respectively. The mode and the standard deviation are 0.018 and 0.00435 s. With these results, it is clear that the convergence time of the 30 tests does not reach any high values, having good behavior according to the standard deviation.

Table 4. Executions performed by SA for a 41 × 41 mesh and a residual value of $\varepsilon = 1 \times 10^{-5}$, for the convection-diffusion problem.

No.	RF_T	t_{conv}	No.	RF_T	t_{conv}
1	0.967	0.013	16	0.928	0.016
2	0.990	0.010	17	0.921	0.016
3	0.975	0.011	18	0.925	0.017
4	0.976	0.012	19	0.930	0.015
5	0.971	0.013	20	0.870	0.022
6	0.880	0.020	21	0.854	0.024
7	0.889	0.019	22	0.860	0.023
8	0.876	0.020	23	0.831	0.026
9	0.897	0.018	24	0.858	0.023
10	0.900	0.018	25	0.859	0.023
11	0.831	0.025	26	0.844	0.025
12	0.880	0.021	27	0.849	0.024
13	0.908	0.018	28	0.910	0.019
14	0.903	0.019	29	0.913	0.018
15	0.910	0.017	30	0.920	0.017

Figure 5 shows the t_{conv} behavior concerning RF_T shown in Table 3. With a leastsquares approximation, it is observed that there exists an inversely proportional relationship, i.e., if RF_T increases, t_{conv} decreases. But many points enclosed by red circles show that with a small decrease in RF_T , t_{conv} tends to increase, which indicates that a directly proportional trend is not followed. It is important to clarify that Figure 5 shows the relaxation factors optimized by the SA algorithm in the 0.83–0.99 interval. The behavior in a range inferior to 0.83 may be different. For example with an RF_T = 0.01, the convergence time in the convection-diffusion problem is 169.13 s, which shows that there is no linear regression behavior in the data.



Figure 5. Behavior of the convergence time t_{conv} of the Convection-Diffusion problem depending on the relaxation factor.

Table 5 shows the convection-diffusion problem's t_{conv} results with the RF_T values found by the SA algorithm. Three tests are carried out for different mesh sizes and a relaxed residual value of $\varepsilon = 1 \times 10^{-5}$. The best values found are compared with the RF_T without optimizing. It is observed that for each mesh size, t_{conv} is improved by more than 60% using the RF_T found by SA, reaching above 90% with a 41 × 41 mesh. By comparing the worst time found by SA (Table 3), which is 0.026 s, in Table 5 is it shown that for a 41 × 1 mesh, t_{conv} continues to improve above 80%. This indicates that any SA solution is efficient in reducing the t_{conv} of the convection-diffusion problem. The *RF* value without optimizing presented in Table 5 is the initial value that SA uses in f(s,RF) of Equation (10). This value is taken to demonstrate that SA can optimize *RF* to reduce the convergence time of the convection-diffusion problem because an *RF* was not found in the literature to make a comparison with the results obtained in this work. Table 5 shows that SA can achieve very good *RF* values that improve the problem's convergence time, starting from a very bad *RF* value. The literature indicates that SA efficiently optimizes its cost function, regardless of the initial solution's quality.

Table 5. Comparison of convergence times obtained from the convection-diffusion problem, applying *RFs* obtained with SA vs. *RF* not optimized, with different mesh sizes and a relaxed residual value of $\varepsilon = 1 \times 10^{-5}$.

Mesh	Without Optimizing <i>RF_T</i>	t_{conv} SA RF_T		t_{conv}	% Improvement
11×11	0.2	0.003	0.66	0.001	67
41×41	0.2	0.137	0.99	0.010	93
61 imes 61	0.2	0.258	0.78	0.027	81

The best three RF_T values found by the SA algorithm are taken from Table 4, and are evaluated with an unrelaxed 61 × 61 mesh and one unrelaxed residual value of $\varepsilon = 1 \times 10^{-10}$. The results are compared with the RF_T without optimizing and they are shown in Table 6, where it is observed that for the first test using the RF_T found by SA, there is a 60% improvement in t_{conv} , while in the remaining two cases there is an improvement of more than 70%.

Table 6. Comparison of convergence times obtained from the convection-diffusion problem, applying the *RFs* obtained with SA vs. *RF* not optimized with residual value of $\varepsilon = 1 \times 10^{-10}$.

Mesh	Without Optimizing <i>RF_T</i>	t _{conv}	SA RF_T	t _{conv}	% Improvement
61×61	0.4	0.208	0.66	0.084	60
61 imes 61	0.4	0.208	0.88	0.034	83
61×61	0.4	0.208	0.78	0.054	74

The average time of the SA algorithm to get the optimal RF_T with the residual value of $\varepsilon = 1 \times 10^{-5}$ and relaxed 41×41 mesh is 0.1135 s. As is shown in Table 6, this time is higher than the t_{conv} obtained using the optimized RF and the unrelaxed residual value of ε $= 1 \times 10^{-10}$ and the unrelaxed 61×61 mesh, but it is less than the t_{conv} used with the RF_T without optimizing. For example, when applying the optimized RF_T of 0.66, the solution converges in 0.085 s. If the average time used to obtain this RF_T is added, the total time applied to solve the problem is 0.1985 s. This result continues to be less than the times obtained when using the RF_T without optimizing presented in Table 6.

4.2. CASE 2: Driven-Cavity Hydrodynamic Problem

Figure 6 presents the mesh independence study. It can be observed that it is possible to work with a 61×61 mesh since its changes are not significant. A larger mesh naturally implies a higher computational cost.



Figure 6. Velocity distribution for different mesh sizes; (a) spatial mesh for *u*. (b) spatial mesh for *v*.

Table 7 shows the Driven-cavity hydrodynamic problem analysis, where the *RFs* help to a more efficient solution convergence. It is observed that there is no clear behavior in the results showing the best values (*RF*) giving a better convergence. The convergence analysis is observed in time where the parameters that help to converge the solution more efficiently to an adequate Δt of time are changed. The obtained errors in the problem convergence are also shown.

	RF			4	Maximal E	Error Found to Ea	to Each Variable		
<i>RF_u</i>	RF_v	RF _p	Δt	l _{conv}	и	υ	p		
0.8	0.8	0.2	0.1	4378.92	$9.99 imes 10^{-11}$	$9.65 imes10^{-11}$	$3.10 imes 10^{-15}$		
0.6	0.4	0.2	0.1	1029.28	$6.68 imes10^{-11}$	$9.99 imes10^{-11}$	$2.53 imes 10^{-13}$		
0.5	0.5	0.1	0.1	864.81	$9.68 imes10^{-11}$	$9.85 imes10^{-11}$	$2.02 imes 10^{-12}$		
0.1	0.1	0.9	0.1	5020.11	$9.99 imes10^{-11}$	$9.66 imes10^{-11}$	$3.10 imes10^{-14}$		
0.1	0.1	0.1	0.1	4289.98	$9.99 imes10^{-11}$	$9.55 imes10^{-11}$	$2.41 imes10^{-15}$		
0.8	0.8	0.2	0.4	177.34	$9.98 imes10^{-11}$	$9.89 imes10^{-11}$	$1.51 imes 10^{-11}$		
0.6	0.4	0.2	0.4	317.04	$6.71 imes10^{-11}$	$9.99 imes10^{-11}$	$6.40 imes10^{-12}$		
0.5	0.5	0.1	0.4	592.36	$4.96 imes10^{-11}$	$5.01 imes10^{-11}$	$9.99 imes10^{-11}$		
0.1	0.1	0.9	0.4	1389.47	$9.96 imes10^{-11}$	$9.65 imes10^{-11}$	$4.35 imes10^{-14}$		
0.1	0.1	0.1	0.4	1256.79	$9.91 imes10^{-11}$	$9.72 imes10^{-11}$	$4.38 imes10^{-12}$		
0.8	0.8	0.2	0.5	134.89	$9.99 imes10^{-11}$	$9.95 imes10^{-11}$	$4.34 imes10^{-11}$		
0.6	0.4	0.2	0.5	258.83	$6.7 imes10^{-11}$	$9.99 imes10^{-11}$	$1.07 imes10^{-11}$		
0.5	0.5	0.1	0.5	701.96	$2.06 imes10^{-11}$	$2.21 imes 10^{-11}$	$9.99 imes10^{-11}$		
0.1	0.1	0.9	0.5	1098.42	$9.9 imes10^{-11}$	$9.6 imes10^{-11}$	$6.5 imes10^{-14}$		
0.1	0.1	0.1	0.5	1256.05	$9.99 imes 10^{-11}$	$9.76 imes 10^{-11}$	$7.67 imes 10^{-12}$		

Table 7. Relaxation factors (*RF*) and Δt variation and errors obtained with a 61 × 61 mesh.

Table 8 shows the tuned values of the SA control parameters for the Driven-Cavity problem.

Table 8. Tuned values of the SA parameters for the driven-cavity problem.

Parameter	Value
T_0	2.00
T_{f}	0.01
MČL	6.00
α	0.95

Table 9 shows the thirty executions of the SA algorithm with an 11×11 mesh and a residual value of $\varepsilon = 1 \times 10^{-5}$. The optimized *RF*s and the convergence time obtained from the Driven-Cavity problem are displayed. The range obtained in the tests concerning

the t_{conv} is 0.016 s, with 0.023, 0.039 and 0.032 s as minimum, maximum, and average times. The mode and standard deviation are 0.035 and 0.00438 s. It is observed that the convergence times of the 30 executions for the driven-cavity problem do not reach any high values and have good behavior according to the standard deviation.

Table 9. Executions performed by SA for an 11×11 mesh and a residual value of $\varepsilon = 1 \times 10^{-5}$, for the driven-cavity problem.

No.	RF_u	RF_v	RF_p	t_{conv}	No.	RF_u	RF_v	RF_p	t_{conv}
1	0.91	0.72	0.24	0.028	16	0.97	0.85	0.27	0.023
2	0.64	0.62	0.22	0.037	17	0.84	0.52	0.23	0.035
3	0.89	0.88	0.43	0.034	18	0.95	0.72	0.23	0.028
4	0.99	0.55	0.46	0.038	19	0.69	0.89	0.22	0.032
5	0.78	0.80	0.44	0.036	20	0.61	0.89	0.21	0.036
6	0.92	0.64	0.61	0.035	21	0.93	0.99	0.54	0.030
7	0.62	0.60	0.22	0.036	22	0.91	0.95	0.55	0.024
8	0.82	0.54	0.32	0.035	23	0.85	0.83	0.56	0.035
9	0.97	0.99	0.53	0.029	24	0.90	0.95	0.28	0.027
10	0.99	0.93	0.55	0.031	25	0.86	0.40	0.41	0.037
11	0.65	0.51	0.19	0.039	26	0.88	0.90	0.28	0.026
12	0.85	0.75	0.17	0.030	27	0.94	0.50	0.37	0.032
13	0.86	0.72	0.17	0.031	28	0.92	0.48	0.38	0.034
14	0.68	0.78	0.23	0.035	29	0.81	0.73	0.25	0.027
15	0.93	0.79	0.24	0.026	30	0.99	0.55	0.34	0.030

Figure 7 shows the t_{conv} behavior concerning the three *RF* presented in Table 9. The convergence time ($t_{conv} \times 10^{-3}$) is shown as a label for each coordinate (RF_u , RF_v and FR_p). It is observed that the t_{conv} presents a slight tendency to decrease when the relaxation factors increase, but this behavior is not entirely proportional as it happens in the first problem. The best t_{conv} appears as a red circle, and a blue circle represents the worst t_{conv} . There are high convergence times with large values in the relaxation factors, and it is observed that the convergence value (yellow circle) that is most repeated (the mode value) is where the relaxation factors have small values and where some have large values. These values indicate that proportionality behavior for this problem cannot be inferred.



Figure 7. Behavior of tconv of the Driven-Cavity problem depending on *RF*s for an 11×11 mesh and a residual value of $\varepsilon = 1 \times 10^{-5}$.

Table 10 shows the t_{conv} values with the relaxation factors found by SA. Three tests are carried out for different mesh sizes and a relaxed residual value of $\varepsilon = 1 \times 10^{-5}$. The best values found by SA are compared with the *RF* without optimizing. It is observed that for each mesh size, t_{conv} is improved by more than 70% using the *RFs* found by SA, reaching above 85% with a 61 × 61 mesh. By comparing the worst time of 0.039 s found by SA, in Table 10 is observed that for an 11 × 11 mesh, t_{conv} improves above 65%. These results

are very similar to those reached with the first case study, and demonstrate that the SA algorithm is an effective approach to reduce the convergence time for this class of problems. The *RF* values without optimizing presented in Table 10 is the initial value that SA uses in f(s,RF) of Equation (10). This value is taken to demonstrate that SA can optimize *RF* to reduce the convergence time of the driven-cavity problems because an *RF* was not found in the literature to make a comparison with the results obtained in this work. Table 10 shows that SA can achieve very good *RF* values that improve the problem's convergence time, starting from a very bad *RF* value. The literature indicates that SA efficiently optimizes its cost function, regardless of the initial solution's quality.

Table 10. Comparison of the convergence times obtained from the driven-cavity problem, applying the relaxation factors obtained with SA vs. *RF* not optimized, with different mesh sizes and a relaxed residual value of $\varepsilon = 1 \times 10^{-5}$.

NC 1	И	ithout	Optimiz	zing	SA				9/ Immercement
Mesh	RF _u	RF_v	<i>RF</i> _p	t_{conv}	<i>RF</i> _u	RF_v	<i>RF_p</i>	t_{conv}	⁷ / ₀ Improvement
11×11	0.2	0.2	0.1	0.113	0.82	0.87	0.27	0.023	80
41×41	0.2	0.2	0.1	2.981	0.43	0.84	0.51	0.694	76
61 imes 61	0.2	0.2	0.1	12.006	0.87	0.58	0.42	1.629	86

Figure 8 presents the behavior of each internal cycle of the SA algorithm concerning the t_{conv} value of the Driven-Cavity problem. For each cycle (Met) with a constant T_{SA} value, a fixed number of solutions are evaluated. It is observed that as the temperature of each cycle is low, the t_{conv} is reduced. When SA reaches its cooling point, the shortest t_{conv} is achieved. For example, a t_{conv} between 470 to 500 s is required by the Met-1 cycle, but between 150 to 200 s by the Met-7 cycle. These values verify that when applying the SA algorithm, it reduced convergence time of the problem, and although short times were evaluated, it is important to remember that the SA tests were carried out for relaxed values of the problem (mesh size and residual value).



Figure 8. Convergence time of the Driven-Cavity problem with SA.

Figure 9 graphically shows the results obtained in Table 8, for the different mesh sizes with the proposed *RFs* and those obtained with the SA algorithm. t_{conv} is compared concerning the number of iterations required to solve the Driven-Cavity problem. It is observed that when comparing the different mesh sizes and relaxation factors, the computational time needed to solve the problem varies. When the *RFs* found by SA are

used, the t_{conv} is improved, decreasing the number of iterations required to obtain a solution for each mesh size. The behavior is opposite when using the *RFs* values without optimizing, since t_{conv} increases considerably, and the number of iterations used, with an unrelaxed 61 × 61 mesh.



Figure 9. Comparison of t_{conv} with respect to the number of iterations for the solution of the Driven-Cavity problem with different relaxation factors (those proposed and those found with SA).

Table 11 presents the results of the Driven-Cavity problem using the *RFs* optimized by the SA algorithm. These factors are used without relaxing the mesh size and the residual value, and the results are compared with those obtained in the problem applying *RFs* without optimizing. The optimized *RFs* offer a significant improvement in convergence, since the computation time is reduced by up to 70% compared to those proposed without optimizing. Even with a very relaxed mesh, such as the 11×11 mesh for the SA tests (Table 9), it is observed that very good convergence times are obtained for the Driven-Cavity problem (Table 11) when the *RFs* shown in Table 9 are used without the relaxed mesh size and residual value.

Table 11. Comparison of convergence times obtained from the Driven-Cavity problem, applying the relaxation factors obtained with SA vs. *RF* not optimized, with mesh and residual unrelaxed values.

Without Optimizing						9	9/ Immercencent		
Mesh	R _u	RF_v	<i>RF</i> _p	t_{conv}	RF _u	RF_v	<i>RF_p</i>	t _{conv}	% Improvement
61×61	0.2	0.2	0.1	528.54	0.82	0.87	0.27	90.08	82
61 imes 61	0.2	0.2	0.1	528.54	0.43	0.84	0.51	103.635	80
61 imes 61	0.2	0.2	0.1	528.54	0.87	0.58	0.42	96.535	81

The average time of the SA algorithm to obtain the three optimal *RFs* with a residual value of $\varepsilon = 1 \times 10^{-5}$ and a relaxed 11×11 mesh is 0.2916 s. As seen in Table 11, this time is less than the t_{conv} obtained using the unrelaxed parameters residual of $\varepsilon = 1 \times 10^{-10}$ and a 61 × 61 mesh. This allows optimization of *RFs*, improving the convergence time of the problem in very short times. For example, when applying the optimized RFs (0.43, 0.84, 0.51), the solution converges at 103.635 s, and the total time to solve the problem is

103.9266 s. This result is less than those obtained with the *RF* values without optimizing presented in Table 11.

Figures 10 and 11, show the results obtained for the Driven-Cavity problem's velocity components concerning the reference [23]. The results shown are obtained based on the relaxation factors optimized by the SA algorithm. The continuous line represents the present work results, and the points in red are those obtained by reference [23]. It is observed that the behavior by the velocities within the problem is very similar. Here u and v are velocities in x and y respectively. The variables because they are in an iterative process represent estimated values, so to denote this the symbol * is added (u^* , v^* , x^* , y^*).



Figure 10. Velocity components within the cavity.



Figure 11. Velocity components within the cavity.

5. Conclusions

It is concluded that the proposed methodology using the SA algorithm is an effective alternative to find the optimized relaxation factors helping to reduce the computational convergence time of the two studied problems, the Convection-Diffusion and the Driven-Cavity problems, since each of the 30 executions of SA present excellent results for each one. This implies that with a single execution of the SA algorithm, well-optimized relaxation factors can be found, reducing the computational convergence time in these problems. The Convection-Diffusion problem results show that the convergence time is improved to more

than 60% using the relaxation factors found by SA, compared with the results obtained using *RF* without optimizing.

For the Driven-Cavity problem, using the best configuration of the three relaxation factors found by the SA algorithm, the computational time improvement is up to 80% compared with the factors without optimizing. With heuristics, it is possible to find suitable factors without the need for expertise, whereas an empirical approach requires expertise on the part of the investigator.

The convergence times in both problems do not show proportional behavior clearly. What they do show is that when a small increase or decrease is made in a relaxation factor, the convergence time can increase or decrease and, in the Driven-Cavity problem that uses three relaxation factors, this conclusion becomes more visible. This convergence behavior based on relaxation factors confirms what is presented in the literature. Therefore, it can be affirmed that the SA algorithm is an effective technique to reduce the computational convergence time of problems involving CFD studies.

It also observed that with a relaxed 11×11 mesh, improved relaxation factors are still obtained to solve the Driven-Cavity problem, and result in reduced computational convergence times than those without relaxing any parameters. This allows SA to find optimized relaxation factors in very short times.

According to the time required by the SA algorithm to obtain the optimized relaxation factors for each of the Convection-Diffusion and Driven-Cavity problems, SA use is a very attractive approach. This is better observed in the Driven-Cavity problem, since the reduction in convergence time added to the time that SA needs to obtain the optimized relaxation factors, allows a very good reduction in convergence time to find the solution.

The present work's contribution is to implement a technique to find the optimal *RF* to improve the convergence of the convection-diffusion and driven-cavity problems. Other authors do not do this and focus more on working with mesh sizes and of Reynolds numbers, or use an under-relaxation scheme with fuzzy logic.

6. Future Work

To reduce the time in which the optimized relaxation factors of the Driven-Cavity problem are obtained, work will be done to parallelize the SA algorithm using parallelization independent of SA, generating multiple executions of SA in parallel as applied in [24], or performing a parallelization of the metropolis cycle as applied in [25]. Either of the two parallelization techniques used will allow a reduction in SA execution time of over 80%, when using 10 central processing units for the execution of the SA algorithm designed to work in parallel. These techniques can be implemented in SA for any type of problem transparently, regardless of whether they are nonlinear or linear problems, with special characteristics such as discrete or continuous optimization.

Author Contributions: Conceptualization, M.A.C.-C. and J.E.-U.; methodology, M.A.C.-C.; software, J.E.-U., R.R.-L. and M.H.C.-R.; validation, Y.L.-N., M.L.E.-D. and M.H.C.-R.; formal analysis, M.H.C.-R., R.R.-L. and M.L.E.-D.; investigation, J.E.-U., Y.L.-N. and R.R.-L.; resources, M.A.C.-C. and J.E.-U.; data curation, R.R.-L. and Y.L.-N.; writing—original draft preparation, M.A.C.-C., J.E.-U. and R.R.-L.; writing—M.A.C.-C. and R.R.-L.; visualization, M.A.C.-C.; supervision, M.A.C.-C. and R.R.-L.; project administration, M.A.C.-C.; funding acquisition, M.M.A.C.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PRODEP, grant number SA-DDI-UAEM/15/451.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Versteeg, H.K.; Malalasekjera, W. An Introduction to Computational Fluid Dynamics-The Finite Volume Method; Prentice: Hoboken, NJ, USA, 2007.
- Xamán, J.; Lira, L.; Arce, J. Analysis of the temperature distribution in a guarded hot plate apparatus for measuring thermal conductivity. *Appl. Therm. Eng.* 2009, 29, 617–623. [CrossRef]
- 3. Crivelli, L.A.; Idelsohn, S.R. Numerical methods in phase-change problems. Arch. Comput. Methods Eng. 1994, 1, 49–74.

- 4. Brandt, A. Multi-Level Adaptive Solutions to Boundary-Value Problem. Math. Comput. 1977, 31, 333–390. [CrossRef]
- Matyka, M. Solution to two-dimensional Incompressible Navier-Stokes Equations with SIMPLE, SIMPLER and Vorticity-Stream Function Approaches. Driven-Lid Cavity Problem: Solution and Visualization. *Comput. Phys. Sect. Theor. Phys.* 2004, arXiv:physics/0407002, 1–13.
- 6. Jafari, A.; Haghighi, A.R. Solution to two-dimensional Incompressible Navier-Stokes Equations with SIMPLE, SIMPLER and Vorticity-Stream Function. *Commun. Adv. Comput. Sci. Appl.* **2015**, 72–82.
- 7. Bonilla, J.; Yebra, L.J.; Dormido, S. A heuristic method to minimise the chattering problem in dynamic mathematical two-phase flow models. *Math. Comput. Model.* **2011**, *54*, 1549–1560. [CrossRef]
- 8. Xamán, J.; Zavala-Guillen, I.; Hernández-López, I.; Uriarte-Flores, J.; Hernández-Pérez, I.; Macías-Melo, E.V.; Aguilar-Castro, K.M. Evaluation of the CPU time for solving the radiative transfer equation with high-order resolution schemes applying the normalized weighting-factor method. *J. Quant. Spectrosc. Radiat. Transf.* **2018**, *208*, 45–63. [CrossRef]
- Xamán, J.; Hernández-López, I.; Uriarte-Flores, J.; Hernández-Pérez, I.; Zavala-Guillen, I.; Moreno-Bernal, P.; Hinojosa, J.F. 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. *Comput. Phys. Commun.* 2018, 231, 72–93. [CrossRef]
- Ryoo, J.; Kaminski, D.; Dragojlovic, Z. Automatic Convergence in a Computational Fluid Dynamics Algorithm Using Fuzzy Logic. In Proceedings of the Conference of the Computational Fluid Dynamics Society of Canada, Halifax, NS, Canada, 30 May–1 June 1999.
- 11. Dragojlovic, Z.; Kaminski, D.A. A Fuzzy Logic algorithm for acceleration of convergence in solving turbulent flow and heat transfer problems. *Numer. Heat Transf. Part B Fundam. Int. J. Comput. Methodol.* **2004**, *46*, 301–327. [CrossRef]
- 12. Cortés, M.; Fazio, P.; Rao, J.; Bustamante, W.; Vera, S. CFD modeling of basic convection cases in enclosed environments: Needs of CFD beginners to acquire skills and confidence on CFD modeling. *Constr. Eng. Mag.* 2014, 29, 22–45.
- 13. Xamán, J.; Gijón-Rivera, M. *Computational Fluid Dynamics for Engineers*; Palibrio: Bloomington, IN, USA, 2016; pp. 29–52, 210–300. (In Spanish)
- 14. Patankar, S.V.; Spalding, D.B. A calculate procedure for heat mass y momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transf.* **1972**, *15*, 1787–1806. [CrossRef]
- 15. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Dover Publications: Mineola, NY, USA, 1998; pp. 150–200.
- 16. Aarts, E.H.L.; Lenstra, J.K. *Local Search in Combinatorial Optimization*, 2nd ed.; Princeton University Press: Princeton, NJ, USA, 2003; pp. 100–150.
- 17. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. Science 1983, 220, 671–680. [CrossRef] [PubMed]
- 18. Johnson, D.S.; Aragon, C.R.; McGeoch, L.A.; Schevon, C. Optimization by Simulated Annealing: En experimental evaluation, Part I, Graph Partitioning. *Oper. Res.* **1989**, *37*, 865–892. [CrossRef]
- 19. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H. Equation of state calculation by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1091. [CrossRef]
- 20. Rayward-Smith, V.J.; Osman, I.H.; Reeves, C.R.; Smith, G.D. *Modern Heuristics Techniques*, 2nd ed.; Wiley: New York, NY, USA, 1996; pp. 1–25.
- Dowsland, K.A.; Díaz, B.A. Heuristic design and fundamentals of the Annealing, Artificial Intelligence. *Iberoam. J. Artif. Intell.* 2003, 7, 93–102.
- 22. Martínez-Oropeza, A. Solution to the Problem of Parallel Machines Not Related by an Ant Colony Algorithm. Master's Thesis, Autonomous University of Morelos State (UAEM), Morelos, Mexico, August 2010. (In Spanish)
- 23. Ghia, U.; Ghia, K.N.; Shin, C.T. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *J. Comput. Phys.* **1982**, *48*, 387–411. [CrossRef]
- 24. Cruz-Chávez, M.A.; Peralta-Abarca, J.D.; Cruz-Rosales, M.H. Cooperative Threads with Effective-Address in Simulated Annealing Algorithm to Job Shop Scheduling Problems. *Appl. Sci.* **2019**, *9*, 3360. [CrossRef]
- Sanvicente-Sánchez, H.; Solís, J.F. A Methodology to Parallel the Temperature Cycle in Simulated Annealing. In *Mexican* International Conference on Artificial Intelligence; Cairó, O., Sucar, L.E., Cantu, F.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1793, ISBN 978-3-540-67354-5.