Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 649630, 18 pages
http://dx.doi.org/10.1155/2014/649630

Research Article

Continuous Genetic Algorithm as a Novel Solver for Stokes and Nonlinear Navier Stokes Problems

Z. S. Abo-Hammour,[1] A. D. Samhouri,[2] and Y. Mubarak[2]

[1] *Mechatronics Engineering Department, Faculty of Engineering and Technology, The University of Jordan, Amman 11942, Jordan*
[2] *Chemical Engineering Department, Faculty of Engineering and Technology, The University of Jordan, Amman 11942, Jordan*

Correspondence should be addressed to Z. S. Abo-Hammour; zaer_hr@yahoo.com

Received 14 March 2014; Accepted 15 May 2014; Published 4 June 2014

Academic Editor: Yang Xu

The one-dimensional continuous genetic algorithm (CGA) previously developed by the principal author is extended and enhanced to deal with two-dimensional spaces in this paper. The enhanced CGA converts the partial differential equations into algebraic equations by replacing the derivatives appearing in the differential equation with their proper finite difference formula in 2D spaces. This optimization methodology is then applied for the solution of steady-state two-dimensional Stokes and nonlinear Navier Stokes problems. The main advantage of using CGA for the solution of partial differential equations is that the algorithm can be applied to linear and nonlinear equations without any modification in its structure. A comparison between the results obtained using the 2D CGA and the known Galerkin finite element method using COMSOL is presented in this paper. The results showed that CGA has an excellent accuracy as compared to other numerical solvers.

1. Introduction

Computational fluid dynamics (CFD), which has been commercialized for few decades, is the science of predicting the effect of different phenomena on the overall performance of the processes through solving sets of mathematical equations that govern these processes using numerical techniques. There are many CFD commercial codes available today, which have been originated by the 80s and 90s, such as Fluent, CFX, Fidap, Polyflow, Phoenix, Star CD, Flow 3D, ESI/CFDRC, and SCRYU [1].

The methodology of CFD is, briefly, applying numerical method (called discretization) to develop approximations of the governing equations of fluid mechanics in the fluid region of interest. This discretization involves the governing differential equations expressed algebraically and the collection of cells (the grid). These sets of equations are solved simultaneously for the flow field variables at each node or cell, which enormously contributes.

The discretized conservation equations are solved iteratively. A number of iterations are usually required to reach a converged solution. The convergence is reached when the changes in solution variables from one iteration to the next are negligible; in other words, the residuals provide a mechanism to help monitor this trend [1]. The major advantage of CFD is the relatively low cost that is continuing to decline as computers and software become more powerful and sophisticated. Moreover, CFD simulations can be executed in a short period of time. It has also the ability of simulating real conditions.

The principle methods currently used in CFD are finite difference method (FDM), finite element method (FEM), the Galerkin method, spectral method (SM), filter scheme methods (FSM), and boundary integral equation methods (BIEM) [1]. The first four methods mentioned above are applicable in principle to all types of fluid flow problems (hyperbolic, parabolic, elliptic, and mixed) whereas (BIEM) method has more restrictions and is usually valuable for elliptic type problems [1].

Fluid flows are encountered in the daily life applications including environmental hazards (air pollution, transport of contaminants, etc.), ventilation and air conditioning applications, combustion in automobile engines, complex flows in furnaces, heat exchangers, chemical reactors, and the human

body processes (blood flow, breathing, drinking, etc.) [2]. These fluid flow applications are expressed mathematically in sets of partial differential equations that represent the conservation principle of mass, momentum, and energy.

The Navier Stokes equations are the basic governing equations for viscous fluid flow. These equations are obtained by applying Newton's law of motion to a fluid element; alternatively these equations are called the momentum equations. These equations are usually supplemented by the mass conservation equation, also called continuity equation. A combination of both momentum equation and continuity equation is usually termed as Navier Stokes equations. The viscous incompressible Navier Stokes equations (INSE) were firmly established in the 19th century as a system of nonlinear partial differential equations that describe the motion of most fluids. As a result, finding a solution with reasonable accuracy of INSE set of equations has attracted the research interest of many scientists around the world [1].

Genetic algorithm (GA) was invented by Holland in the 1960s at the University of Michigan [3]. In contrast to evolution strategies and programming, Holland's original objective was not to design algorithms that solve specific problems but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems [4].

GA was recently reviewed by Goldberg [5], Davis [6], and others. In general, GA is implemented using computer simulations in which an optimization problem is specified. A genetic algorithm begins with a population of typically random parents in which members of a space of candidate solutions called individuals are represented using abstract representations called chromosomes [4]. In each generation, multiple individuals are randomly selected from the current population based upon some application of fitness and then using crossover and modified through mutation to form a new population. And finally it reaches the fit population with the exact one.

There is no rigorous definition of "genetic algorithm" accepted by all in the evolutionary computation community that differentiates GA from other evolutionary computation methods. However, it can be said that most methods called "GA" have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring [7].

Continuous genetic algorithms (CGAs) were developed by Abo-Hammour as an efficient method for finding the global solutions of smooth functions [7]. In CGAs, the smoothness of the solution curve is achieved. Their novel development has opened a wide venue for different engineering and mathematical applications. In the numerical field, CGAs have been applied for the numerical solution of boundary value problems [8–13], Laplace equations [14], differential-algebraic systems [15], fuzzy differential equations [16], and the solution of Stokes and nonlinear Navier Stokes problems [17]. CGAs have been also applied in robotics [18, 19] and optimal control problems [20, 21].

When using GA in optimization problems, there are two vital points that should be considered: firstly, seeking the interrelation between parameters of optimization and secondly whether there is some restriction on the smoothness of the resulting curve or not [8]. In the case of uncorrelated parameters or nonsmooth solution curves, the conventional genetic algorithm (GA) performs well and there is no need of the CGA. On the other hand, if the parameters are interrelated or if the smoothness of the solution curve is a must, then the continuous genetic algorithm (CGA) is preferable [18].

In summary, CGA has many advantages compared with conventional GA especially when it is applied to problems where smooth solution curve is a need and when more than one parameter is used [7].

(1) CGA needs smaller memory than what GA needs, because GA uses genotype and phenotype representations of the population's individuals while CGA utilizes only the phenotype data. This makes CGA more suitable to problems with larger number of parameters.

(2) Execution time for CGA is smaller than time needed by GA because there are no encoding/decoding processes in CGA.

(3) The conventional GA cannot be used in applications where the optimal solution is required.

The novel application of the (CGA) possesses several advantages: (i) it guarantees the smoothness of the solution curves, (ii) the results obtained using CGA are found to be in good agreement with the analytical solutions, and (iii) it can be applied to linear and nonlinear problems without any modification in the algorithm.

In this paper, the novel method based on continuous genetic algorithms is applied to solve two problems modeled by the steady state Navier Stokes equations in two-dimensional rectangular regions [17]. These problems varied in terms of linearity, that is, linear and nonlinear problems of Dirichlet boundary conditions. The main objective of this work is to solve algebraic equations that result from finite differencing of the partial differential equations (PDE) by CGA as a global optimization technique. The efficiency of CGA method is demonstrated by comparing its results to the results of other well-known methods. Furthermore, the exact solutions for both cases are already known, which make the judgment of the CGA method results possible. The usage of exact solutions here is only to reveal the efficiency of CGA in solving problems.

2. Theory of Continuous Genetic Algorithm

Continuous genetic algorithms (CGAs) were firstly developed to avoid the sharp jumps in the parameter values which can be resulted by using conventional genetic algorithm (GA) form in some optimization problems. That can be shown when the parameters of the optimization problem

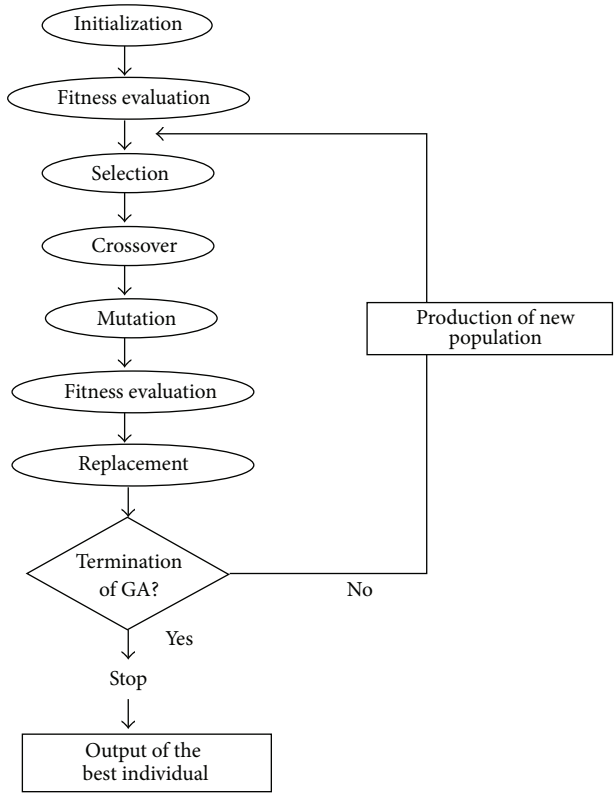

Figure 1: Block diagram for GA procedure.

are correlated with each other or if the smoothness of the solution curve is a must [8]. Continuous genetic algorithm (CGA) mainly depends on the evolution of curves in one-dimensional space, surfaces in two-dimensional space, and volumes in three-dimensional space [18].

The CGAs are of global nature; consequently, the operators of CGA result in smooth transitions but the results of the traditional GA are a step-function-like jump in the parameter values [8]. As discussed by Abo-Hammour et al. [14], the solution of two-dimensional equation in the Cartesian coordinates was found using continuous genetic algorithm (CGA). Following Laplace equation other equations such as Stokes type can be discretized in the same way [8].

The central difference discretization form of two-dimensional equations is

$$u_{i,j} = g\left(u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}\right). \tag{1}$$

The nodal residue for node (i, j) is defined as

$$r_{i,j} = u_{i,j} - g\left(u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}\right). \tag{2}$$

The overall residue for equation is the sum of squares of the nodal residues and is stated as

$$R = \sqrt{\sum_{i=1}^{n_x} \sum_{j=1}^{m_y} r_{i,j}^2}, \tag{3}$$

where n_x is the number of unknown nodes along the x-direction and m_y is the number of unknown nodes along the y-direction.

Then, an appropriate fitness function can be given as [7]

$$F = \frac{1}{1+R}. \tag{4}$$

This means that the larger the residues value the smaller the fitness one. For the exact solution $R \approx 0$ and $F \approx 1$.

The CGAs consist of general steps as discussed by [8] and are followed during solving problems and these steps are shown in Figure 1.

(1) Initialization. This step is just like an initial guess for the solution curve that modified functions can be used to randomly generate the initial population or parents [22]. In general, the initialization functions used in the algorithm should satisfy the following conditions.

(1) It should be smooth from one side and should satisfy any problem-specific constraints, if such constraints exist, from the other side [7]. That is, any smooth function that is close enough to the expected solution curve and satisfies the boundary values can be used.

(2) It should result in an information-rich initial population based on the population diversity, which is split into two main types.

(a) *Smooth Functions Diversity*. This is the diversity due to using a mixture of smooth functions in the initialization phase instead of one function. The expected solution curves are not known and correspondingly a mixture of smooth functions that satisfy the given boundary values will be beneficial to result in a diverse initial population.

(b) *Parameters Diversity*. This is the diversity due to the random generation of the initialization parameters, and in order to make the initial population as much diverse as we can, randomness should be there to remove any bias toward any solution.

It is to be noted that the closer the initialization function to the final solution, the faster the convergence speed [8–13]. However, this convergence speed improvement is minor. This means that the initialization functions have minor effect on the convergence speed of the algorithm because usually the effect of the initial population dies after few tens of generations and the convergence speed after that is governed by the selection mechanism, crossover, and mutation operators.

It is also worth mentioning that the population size is kept constant throughout the evolution process of the algorithm. This population size affects the convergence speed, the average fitness, and the corresponding errors of CGA [7–15]. Small population sizes suffer from larger number of generations required for convergence and the probability of being trapped in local minima, while large population sizes suffer from larger number of fitness evaluations that means larger execution time. Previous studies and analysis performed showed that a compromise population size for similar problems with similar number of unknown nodes is about 500 individuals [7–12].

The initial population in this case will be divided into four equal segments; each of them generated using a different set of smooth functions to obtain a diverse initial population. The first part of the initial population is generated using two-dimensional Gaussian functions, $n_1(x, y)$, the second segment of the initial population is generated using a correlated two-dimensional tangent hyperbolic function, $n_2(x, y)$, and the third part $n_3(x, y)$ is generated using an uncorrelated two-dimensional tangent hyperbolic, while the last segment $n_4(x, y)$ is generated using a mixture of the above three functions. The four functions are given as follows [22]:

$$
\begin{aligned}
&n_1(x, y) \\
&\quad = \sum_{i=1}^{k} B_{1,i} \\
&\qquad \times \exp\left[-\frac{1}{2\left(1 - l_i^2\right)} \right. \\
&\qquad\quad \times \left\{ \frac{(x - c_{x,i})^2}{\sigma_{x,i}^2} \right. \\
&\qquad\qquad \left. \left. -2l_i \frac{(x - c_{x,i})(y - c_{y,i})}{\sigma_{x,i}\sigma_{y,i}} + \frac{(y - c_{y,i})^2}{\sigma_{y,i}^2} \right\} \right], \\
&n_2(x, y) \\
&\quad = \sum_{i=1}^{k} B_{2,i} \times 0.5 \\
&\qquad \times \left(1 + \text{hyp tan} \left[\frac{q_{x,i} \times (x - c_{x,i})}{\sigma_{x,i}} \right. \right. \\
&\qquad\quad + 2l_i \frac{(x - c_{x,i})(y - c_{y,i})}{\sigma_{x,i}\sigma_{y,i}} \\
&\qquad\quad \left. \left. + \frac{q_{y,i}(y - c_{y,i})}{\sigma_{y,i}} \right] \right), \\
&n_3(x, y) = B_3 + 0.5 \\
&\qquad \times \left(1 + \text{hyp tan} \left\{ \frac{q_x (x - c_x)}{\sigma_x} \right\} \right. \\
&\qquad\quad \left. \times \text{hyp tan} \left\{ \frac{q_y (y - c_y)}{\sigma_y} \right\} \right), \\
&n_4(x, y) = n_1(x, y) + n_2(x, y) + n_3(x, y),
\end{aligned}
\tag{5}
\tag{6}
$$

where k is a random in the range $[1, N/5]$ and N is the number of unknown nodes in the grid. B_1, B_2 are a random number in the range $[-2, 2]$ and B_3 is a random number in the range $[-0.25, 0.25]$. c_x is a random number in the range $[x_{\min}, x_{\max}]$. σ_x is a random number in the range $[0, x_{\max} - x_{\min}]$. c_y is a random number in the range $[y_{\min}, y_{\max}]$. σ_y is a random number in the range $[0, y_{\max} - y_{\min}]$. l is a random number in the range $[-0.5, 0.5]$. q_x, q_y are random numbers in the range $[0, 2]$ and in $n_4(x, y)$, $k = 1$, for both n_1 and n_2.

The upper functions $n(x, y)$ are then rescaled within the range $[I_{\min}, I_{\max}]$ representing the expected minimum and maximum initial nodal values.

It is to be noted that the parameter "l" appearing in $n_1(x, y)$ and $n_2(x, y)$ as given in (5) is chosen between -0.5 and $+0.5$ based on the second term of the argument of the exponential or the tangent hyperbolic function which is related to the dual effect of both coordinates (x, y). The

coefficient of that term is "2l" and in order to have a contribution for that term within the range $[-1, 1]$, then the range of the "l" values should be within the range $[-0.5, 0.5]$.

(2) Fitness Evaluation. In this step, the measure of goodness of each individual in the population is calculated. Fitness will be applied to all populations until reaching the maximum criteria which is around one. This step is applied according to (2)–(4) [7].

(3) Selection. Selection process is to choose the individuals according to their relative fitness in order to enter the devotion process to create the new individuals for the next generation. This step ensures that the overall quality of the population increases from one generation to the next [7].

(4) Crossover. Crossover is a major operation that really empowers the GA. It operates on two randomly selected highly fitted individuals at a time and generates offspring by combining both parent individuals' features [14].

In two-dimensional cases, crossover function will also be expanded into the two-dimensional following form [14]:

$$O_1(x,y) = S(x,y) \times I_1(x,y) + (1 - S(x,y)) \times I_2(x,y), \tag{7a}$$

$$O_2(x,y) = (1 - S(x,y)) \times I_1(x,y) + S(x,y) \times I_2(x,y), \tag{7b}$$

where O_1, O_2 are the two children resulted from the crossover of the two parents I_1 and I_2. S is the mixing function in the range $[0, 1]$.

Four types of mixing function can be used: the first three types are tangent hyperbolic-related functions, while the last one is of Gaussian shape [14]:

$$S_1(x,y) = 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x - c_x)}{\sigma_x} + \frac{q_y(y - c_y)}{\sigma_y}\right]\right),$$

$$S_2(x,y) = 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x - c_x)}{\sigma_x} + \frac{q_{x,y}(x - c_x)(y - c_y)}{\sigma_x\sigma_y} + \frac{q_y(y - c_y)}{\sigma_y}\right]\right),$$

$$S_3(x,y) = 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x - c_x)}{\sigma_x}\right] \times \text{hyp tan}\left[\frac{q_y(y - c_y)}{\sigma_y}\right]\right),$$

$$S_4(x,y) = \sum_{i=1}^{N_c} \exp\left[-\frac{1}{2(1 - l_i^2)} \times \left\{\frac{(x - c_{x,i})^2}{\sigma_{x,i}^2} - 2l_i\frac{(x - c_{x,i})(y - c_{y,i})}{\sigma_{x,i}\sigma_{y,i}} + \frac{(y - c_{y,i})^2}{\sigma_{y,i}^2}\right\}\right], \tag{8}$$

where

$$c_x = [x_{\min} + 0.1 \times (x_{\max} - x_{\min}), 0.9 \times (x_{\max} - x_{\min})],$$
$$c_y = [y_{\min} + 0.1 \times (y_{\max} - y_{\min}), 0.9 \times (y_{\max} - y_{\min})]. \tag{9}$$

σ_x, σ_y are chosen such that the tangent hyperbolic function achieves its complete transition from –1 to 1 within the given ranges of both x and y. q_x, q_y have values +1, –1 randomly. $q_{x,y}$ is random number in the range $[-0.2, 0.2]$. N_c represents the number of crossing points and it is chosen in the range $[1, N/20]$. l is a random number within the range $[-0.1, 0.1]$.

(5) Mutation. Mutation introduces random variations into the population. It is applied to a single chromosome only. It is usually performed with low probability. Genetic diversity of the population must be maintained so in order to guard against premature convergence, mutation is usually taken into consideration [14].

Two-dimensional mutation processes can be governed by the following formula [14]:

$$O^m(x,y) = O(x,y) + w \times M(x,y), \tag{10}$$

where $O(x, y)$ represents the child produced from the crossover process. $O^m(x, y)$ is the mutated child. M is the mutation function. w is a random number in the range

$[-R_{\text{ave}}, R_{\text{ave}}]$ and R_{ave} is average overall residue of the population. There are three types of mutation functions [7, 8]:

$$\begin{aligned}
&M_1(x,y) \\
&\quad = m_{\text{dc1}} \\
&\qquad + \exp\left[-\frac{1}{2(1-l_i)}\right. \\
&\qquad\quad \times \left\{\frac{(x-c_x)^2}{\sigma_x^2}\right. \\
&\qquad\qquad \left.\left. - 2l\frac{(x-c_x)(y-c_y)}{\sigma_x\sigma_y} + \frac{(y-c_y)^2}{\sigma_y^2}\right\}\right], \\
&M_2(x,y) \\
&\quad = m_{\text{dc2}} + 0.5 \\
&\qquad \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x}\right.\right. \\
&\qquad\qquad + \frac{q_{x,y}(x-c_x)(y-c_y)}{\sigma_x\sigma_y} \\
&\qquad\qquad \left.\left. + \frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&\qquad M_3(x,y) = m_{\text{dc3}},
\end{aligned} \tag{11}$$

where c_x is a random number in the range $[x_{\min}, x_{\max}]$. σ_x is a random number in the range $[0, x_{\max} - x_{\min}]$. c_y is a random number in the range $[y_{\min}, y_{\max}]$. σ_y is a random number in the range $[0, y_{\max} - y_{\min}]$. q_x, q_y have values +1, −1 randomly. $q_{x,y}$ is random number in the range $[-0.2, 0.2]$. $m_{\text{dc1}}, m_{\text{dc2}}$ are some random dc offset values within the range $[-0.25, 0.75]$. l is a random number within the range $[0, 1]$. m_{dc3} is a random number within the range $[-0.5, 0.5]$.

There are three methods for selecting the value of c that is usually used in the mutation process [8].

(1) Random method: in this method c is randomly selected as given in the crossover step.

(2) Lamarckian method: this method depends on the value of residuals at certain mesh point that is the larger residual value at that point then the probability of choosing this point as a mutation center.

(3) Deterministic manner: in this method the mesh point with the maximum residual is chosen as the mutation center.

CGA also used "extinction and immigration" operator. This operator is applied when all individuals in the population are identical or when the improvement in the fitness value of the best individual over a certain number of generations is less than some threshold value. This means that no new information will be obtained through crossover process [7]. The CGA thus tends to stagnate; "extinction and immigration" operator is used to bypass this difficulty. This operator, as indicated by its name, consists of two stages; the first stage is the extinction process where all of the individuals in the current generation are removed except the best of generation individual [7]. The second stage is the mass-immigration process, where the extinct population is filled out again by generating $N_p - 1$ individuals to keep the population size fixed. The generated population is divided into two equal segments each of $N_p/2$ size; the first segment, with $2 \le j \le N_p/2$, is generated as in the initialization phase, while the other segment is generated by performing continuous mutations to the best-of-generation individual as given by the following formula [14]:

$$O_i(x,y) = O_1(x,y) + F \times G(x,y), \quad 2 \le i \le \frac{N_p}{2}, \tag{12}$$

where O_i is the ith parent generated using immigration operator. O_1 is the best of generation individual. F is a random number in the range $[-R_1, R_1]$, where R_1 is the overall residue of the best individual and $G(x, y)$ is one of the following functions [14]:

$$\begin{aligned}
&G_1(x,y) \\
&\quad = m_{\text{dc}} + 0.5 \\
&\qquad \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x}\right.\right. \\
&\qquad\qquad + \frac{q_{x,y}(x-c_x)(y-c_y)}{\sigma_x\sigma_y} \\
&\qquad\qquad \left.\left. + \frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&G_2(x,y) \\
&\quad = m_{\text{dc}} + 0.5 \\
&\qquad \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x}\right]\right. \\
&\qquad\quad \left. \times \text{hyp tan}\left[\frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&G_3(x,y) \\
&\quad = m_{\text{dc}}
\end{aligned}$$

$$+\exp\left[-\frac{1}{2\left(1-l_i\right)}\times\left\{\frac{\left(x-c_x\right)^2}{\sigma_x^2}-2l\frac{\left(x-c_x\right)\left(y-c_y\right)}{\sigma_x\sigma_y}+\frac{\left(y-c_y\right)^2}{\sigma_y^2}\right\}\right]$$
$$G_4\left(x,y\right)=m_{\mathrm{dc4}}. \tag{13}$$

q_x, q_y have values +1, −1 randomly. $q_{x,y}$ is random number in the range $[-1,1]$. m_{dc} are some random dc offset values within the range $[-0.25, 0.75]$. l is a random number within the range $[-0.5, 0.5]$. m_{dc4} is a random number within the range $[0,1]$.

(6) Replacement. Applying the genetic operators to the parent's population generates the offspring's population and then the parents' population is totally replaced by the offspring's population. This is called nonoverlapping, generational, or replacement. In this step the "life cycle" of the population can be completed [7].

(7) Termination. When some convergence criterion is met, the GA is terminated. The convergence criteria are as follows: (i) the fitness of the best individual so far found exceeds a threshold value, (ii) the maximum number of generations is reached or the progress limit, and (iii) the improvement in the fitness value of the best member of the population over a specified number of generations is less than some predefined threshold. The best solution of the problem is found after the termination of the algorithm [7].

3. Numerical Results for Stokes Equations

The general forms of Stokes equations are shown in the following:

$$-\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]=f_x,$$
$$-\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]=f_y, \tag{14}$$

in which the right hand side is the shear stress term and the left hand side is the external force applied on the fluid, with continuity equation

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0. \tag{15}$$

Dennis and Hudson [23] tried to solve the nonlinear form of Navier Stokes equations but with no pressure term as shown in the following equations:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]+f_x, \tag{16}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]+f_y. \tag{17}$$

Continuity equation is as follows:

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0, \tag{18}$$

within the range of $x=[0,1]$ and $y=[0,1]$.

Wang et al. [24] tried to solve Stokes equation using finite element method within the region $x=[0,1]$, $y=[0,1]$; the authors used the following boundary conditions:

$$\begin{aligned} u\left(0,y\right)&=0, & u\left(x,0\right)&=0,\\ u\left(1,y\right)&=0, & u\left(x,1\right)&=0,\\ v\left(0,y\right)&=0, & v\left(x,0\right)&=0,\\ v\left(1,y\right)&=0, & v\left(x,1\right)&=0. \end{aligned} \tag{19}$$

The exact solutions as given by Wang et al. [24] are as follows:

$$\begin{aligned} u&=-2xy\left(x-1\right)\left(y-1\right)x\left(x-1\right)\left(2y-1\right),\\ v&=2xy\left(x-1\right)\left(y-1\right)y\left(y-1\right)\left(2x-1\right). \end{aligned} \tag{20}$$

The right hand sides for (14) were given by Wang et al. [24] in the following forms:

$$\begin{aligned} f_x=&\,8x^2(x-1)^2\left(y-1\right)+4x^2\left(2y-1\right)(x-1)^2\\ &+8x^2y(x-1)^2+4y\left(2y-1\right)(x-1)^2\left(y-1\right)\\ &+4x^2y\left(2y-1\right)\left(y-1\right)\\ &+8xy\left(2x-2\right)\left(2y-1\right)\left(y-1\right),\\ f_y=&-8y^2\left(x-1\right)\left(y-1\right)^2-4y^2\left(2x-1\right)\left(y-1\right)^2\\ &-8xy^2\left(y-1\right)^2-4x\left(2x-1\right)\left(x-1\right)\left(y-1\right)^2\\ &-4xy^2\left(2x-1\right)\left(x-1\right)\\ &-8xy\left(2x-1\right)\left(2y-2\right)\left(x-1\right). \end{aligned} \tag{21}$$

This right hand sides are resulted by substituting the exact solutions shown in (20) into (14). The nodal values of the exact solutions are shown in Tables 1 and 2.

Moreover, the graphical representations of those actual values are shown in Figures 3(a) and 3(b).

The boundary values are known from the main problems but the other nodal values are unknown values of u and must

Table 1: Actual u nodal values for Stokes equations including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0055	−0.006	−0.0055	−0.0042	−0.0025	−0.0008	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	0.00078	0.00246	0.00423	0.0055	0.006	0.00553	0.0042	0.0025	0.00078	0
y_7	0	0.00136	0.0043	0.00741	0.0097	0.0105	0.00968	0.0074	0.0043	0.00136	0
y_8	0	0.00156	0.00492	0.00847	0.0111	0.012	0.01106	0.0085	0.0049	0.00156	0
y_9	0	0.00117	0.00369	0.00635	0.0083	0.009	0.00829	0.0064	0.0037	0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 2: Actual v nodal values for Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.001166	0.001555	0.001361	0.000778	0	−0.00078	−0.00136	−0.00156	−0.00117	0
y_2	0	0.003686	0.004915	0.004301	0.002458	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_3	0	0.00635	0.008467	0.007409	0.004234	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_4	0	0.008294	0.011059	0.009677	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.008294	0.011059	0.009677	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_7	0	0.00635	0.008467	0.007409	0.004234	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_8	0	0.003686	0.004915	0.004301	0.002458	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_9	0	0.001166	0.001555	0.001361	0.000778	0	−0.00078	−0.00136	−0.00156	−0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

be solved. These nodes can be represented in the (10×10) mesh of 100 elements, as shown in Figure 2.

With the same previous conditions COMSOL solution was obtained and the nodal values of u and v variables are shown in Tables 3 and 4. Moreover, the graphical representations for u and v COMSOL solutions are shown in Figures 3(c) and 3(d).

Appling (22) and using the errors between the exact and COMSOL values, it is found that L_2 norm error equals to 1.29×10^{-5} for u variable and about 1.21×10^{-5} for v variable with average time for solving the problem of about 0.045:

$$L_2 \text{ norm error} = \sqrt{\sum_{i=1}^{n} \left\| u_{\text{exact}} - u_{\text{numerical}} \right\|^2}. \quad (22)$$

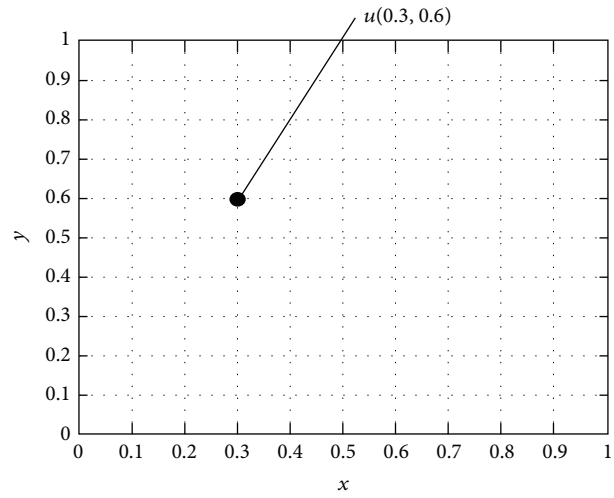

Figure 2: Rectangular mesh with step size equal 0.1.

3.1. Solution of Stokes Equations Using Continuous Genetic Algorithm. All steps of CGA are applied here and they are repeated till the convergence criterion is met (fitness = 0.99), where the best unknown values of nodes are reached. The central difference discretization form of Stokes equation was shown in (23) but for both u and v separately; also (24) and

Table 3: Nodal values of u variable using COMSOL for Stokes case solution.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0055	−0.006	−0.0055	−0.0042	−0.0025	-7.78×10^{-4}	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	7.79×10^{-4}	0.0025	0.0042	0.0055	0.006	0.0055	0.0042	0.0025	7.78×10^{-4}	0
y_7	0	0.0014	0.0043	0.0074	0.0097	0.0105	0.0097	0.0074	0.0043	0.0014	0
y_8	0	0.0016	0.0049	0.0085	0.0111	0.012	0.0111	0.0085	0.0049	0.0016	0
y_9	0	0.0012	0.0037	0.0064	0.0083	0.009	0.0083	0.0063	0.0037	0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 4: Nodal values of v variable using COMSOL for Stokes case solution.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00117	0.00156	0.00136	7.79×10^{-4}	0	-7.79×10^{-4}	−0.00136	−0.00156	−0.00117	0
y_2	0	0.00369	0.00492	0.0043	0.00246	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_3	0	0.00635	0.00847	0.00741	0.00423	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_4	0	0.00829	0.01106	0.00967	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.00829	0.01106	0.00967	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_7	0	0.00635	0.00847	0.00741	0.00423	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_8	0	0.00369	0.00492	0.0043	0.00246	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_9	0	0.00117	0.00156	0.00136	7.79×10^{-4}	0	-7.79×10^{-4}	−0.00136	−0.00156	−0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 5: Nodal u values including boundaries for Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0056	−0.006	−0.0056	−0.0043	−0.0025	−0.0008	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	0.0008	0.0024	0.0042	0.0055	0.006	0.0055	0.0042	0.0024	0.0008	0
y_7	0	0.0013	0.0043	0.0074	0.0096	0.0105	0.0097	0.0074	0.0043	0.0014	0
y_8	0	0.0015	0.0049	0.0084	0.011	0.012	0.011	0.0085	0.0049	0.0016	0
y_9	0	0.0012	0.0037	0.0063	0.0083	0.009	0.0083	0.0064	0.0037	0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

(25) are used in order to define the unknown residual nodal u and v values:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f_x = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{h^2} + f_x, \tag{23}$$

where h is the step size in which h_x is the step size in the x-direction and h_y is the step size in the y-direction. And in this case $h_x = 0.1$ and $h_y = 0.1$, again the problem is defined over the rectangular region $x = [0, 1]$, $y = [0, 1]$.

Then, after rearrangement, one has

$$u_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} + \frac{f_x}{4} \times h^2. \tag{24}$$

The nodal residue for node (i, j) is defined as

$$r_{i,j} = u_{i,j} - \left[u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} + \frac{f_x}{4} \times h^2 \right]. \tag{25}$$

Table 6: Nodal v values including boundaries for Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.0012	0.0015	0.0013	0.0008	0	−0.0008	−0.0014	−0.0016	−0.0012	0
y_2	0	0.0037	0.0049	0.0043	0.0024	0	−0.0025	−0.0043	−0.0049	−0.0037	0
y_3	0	0.0063	0.0085	0.0074	0.0042	0	−0.0042	−0.0074	−0.0085	−0.0064	0
y_4	0	0.0083	0.011	0.0097	0.0055	0	−0.0055	−0.0097	−0.0111	−0.0083	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.0083	0.0111	0.0097	0.0055	0	−0.0055	−0.0097	−0.0111	−0.0083	0
y_7	0	0.0063	0.0085	0.0074	0.0042	0	−0.0042	−0.0074	−0.0085	−0.0064	0
y_8	0	0.0037	0.0049	0.0043	0.0025	0	−0.0025	−0.0043	−0.0049	−0.0037	0
y_9	0	0.0011	0.0016	0.0014	0.0008	0	−0.0008	−0.0014	−0.0016	−0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 7: Residual u values for all nodes for Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	3.9746	1.6466	6.6173	4.3473	9.92845	0.3673	4.8303	2.7018	14.821	0
y_2	0	1.4584	9.9158	7.0978	2.2746	0.28946	3.0963	7.1133	8.0372	2.23402	0
y_3	0	6.9231	6.3902	5.4447	0.828	14.9252	4.7176	18.645	0.7471	12.2662	0
y_4	0	14.162	3.6126	13.076	11.32	9.05015	11.809	14.497	9.7418	0.92215	0
y_5	0	1.1706	0.4675	10.523	7.319	6.62552	3.2928	14.653	7.4677	0.74626	0
y_6	0	3.9953	0.8315	4.3266	2.8571	5.79388	12.773	2.5444	8.3358	0.7428	0
y_7	0	4.9107	4.7092	20.41	10.882	11.2205	8.4874	10.567	4.3102	5.81061	0
y_8	0	6.3028	1.5624	14.346	0.5647	0.08398	7.2523	7.6049	0.0954	2.45801	0
y_9	0	10.577	2.1018	5.8802	0.9785	1.1455	3.3996	3.2041	4.0743	1.87985	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 8: Residual v values for all nodes for Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	2.1016	8.8962	8.8272	5.753	5.4979	16.1784	12.78	0.0776	6.7879	0
y_2	0	5.1713	1.5391	2.8585	2.7778	4.2006	4.71092	4.7172	6.9823	5.42282	0
y_3	0	7.7604	6.4171	11.901	13.982	5.4556	15.6543	13.987	1.2369	14.1758	0
y_4	0	9.7749	8.5258	13.076	7.6596	7.2225	8.32927	8.4301	1.9981	13.6789	0
y_5	0	0.532	3.1775	0.0507	3.1278	8.7859	4.20619	4.328	2.846	5.58614	0
y_6	0	14.067	1.6875	15.331	0.6956	13.451	4.39109	11.482	6.6231	0.66414	0
y_7	0	8.3609	2.5554	6.8751	21.146	2.2431	11.059	11.517	8.1391	3.99419	0
y_8	0	8.1196	9.4279	12.997	0.6023	0.3942	4.97137	5.9129	5.6786	1.88925	0
y_9	0	0.793	2.9236	5.7456	4.52	5.8774	1.37201	0.4039	2.9202	1.80238	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Then, the overall residues were solved for (25) as the sum of squares of the nodal residues, as stated in (3); after that the appropriate fitness function resulted from (4).

The obtained solution for Stokes equation [Dirichlet boundary conditions] using continuous genetic algorithms (CGAs) are shown in Tables 5 and 6, including the nodal values of the boundaries which are equal to zero. And the graphical representations of these values are shown in Figures 3(e) and 3(f).

By calculating the errors between the actual nodal values and CGA one, the L_2 norm error is found to be equal 3.02×10^{-4} for u variable and 1.08×10^{-4} for v variable. The time needed to reach solution in CGA was about 7523 seconds.

Wang et al. [24] used finite element method with a mesh size of 8×8 for solving Stokes equations and then reported 5.05×10^{-4} as L_2 norm error of the solution.

Comparing the error values which resulted by applying CGA with the errors given by finite element method by Wang,

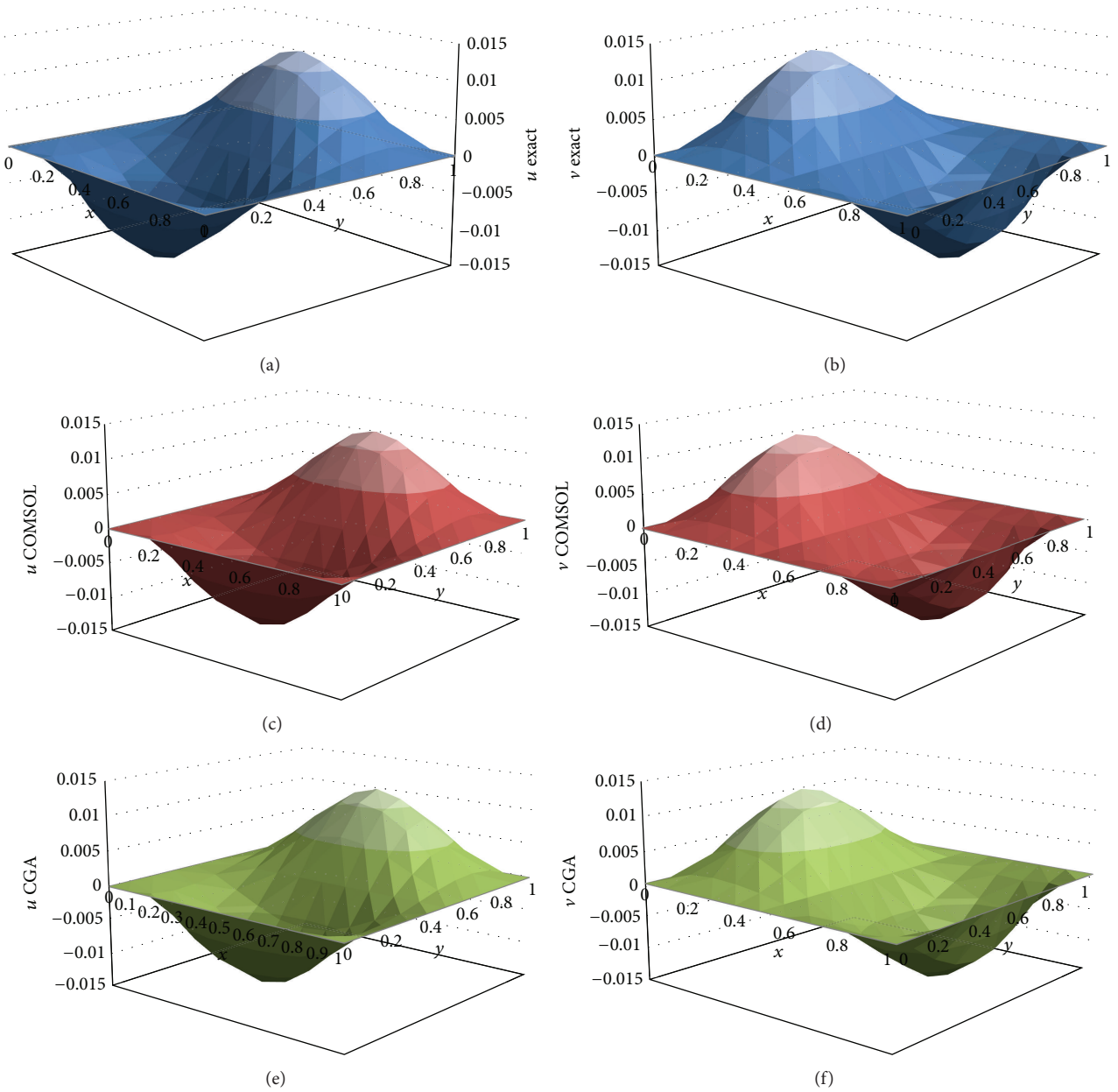

FIGURE 3: For Stokes case. (a) Graphical representation for actual values of u. (b) Graphical representation for actual values of v. (c) Finite element solution for u variable using COMSOL. (d) Finite element solution for v variable using COMSOL. (e) Graphical representation for nodal u values using CGA. (f) Graphical representation for nodal v values using CGA.

it is found that the error for solution obtained by CGA is close to that error for solution obtained by finite element. On the other hand, L_2 norm error for using COMSOL is less, that was about 1.29×10^{-5} for u variable and about 1.21×10^{-5} for v values and even these errors are less than the errors of CGAs method.

The average number of generations was 1139, and the average residual value was about 6.339×10^{-4}. Then, the average fitness according to (4) is very close to the maximum fitness value of 1.

The convergence state of nodal values of the nodes was presented along all generations such as an example for $[u(2,2), u(6,6)$, and $u(6,10)]$ nodes excluding the boundary values as shown in Figures 4, 5, and 6, respectively. It is concluded from the figures that steady state is reached for $u(2,2)$ nodes much earlier than that for $u(6,6)$ and $u(6,10)$. The reason for that is that the node $(2,2)$ location is closer to the boundaries compared with the other nodes locations. For the same reason, the convergence for node $(6,10)$ is faster than for $(6,6)$.

The nodal residues in this case were less than 14.925×10^{-4} for u values as shown in Table 7 and were less than 18.441×10^{-4} for v values as shown in Table 8. It can be concluded from these values that the solution is close to the exact solution.

Table 9: Actual u nodal values for nonlinear Navier Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
y_2	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2
y_3	0	0.03	0.06	0.09	0.12	0.15	0.18	0.21	0.24	0.27	0.3
y_4	0	0.04	0.08	0.12	0.16	0.2	0.24	0.28	0.32	0.36	0.4
y_5	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
y_6	0	0.06	0.12	0.18	0.24	0.3	0.36	0.42	0.48	0.54	0.6
y_7	0	0.07	0.14	0.21	0.28	0.35	0.42	0.49	0.56	0.63	0.7
y_8	0	0.08	0.16	0.24	0.32	0.4	0.48	0.56	0.64	0.72	0.8
y_9	0	0.09	0.18	0.27	0.36	0.45	0.54	0.63	0.72	0.81	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

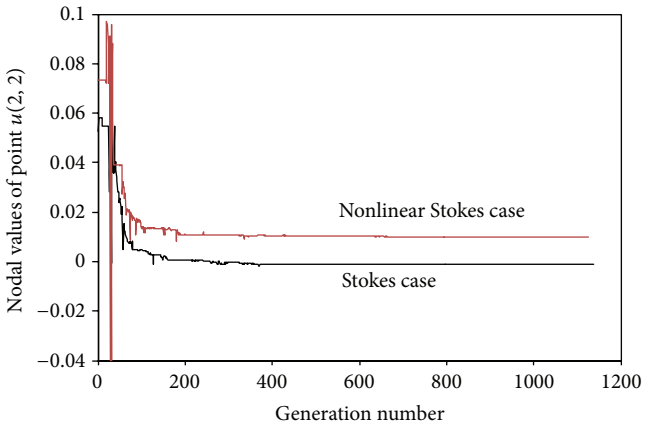

Figure 4: Nodal values of the point $u(2, 2)$ along all generations for Stokes and nonlinear Stokes cases.

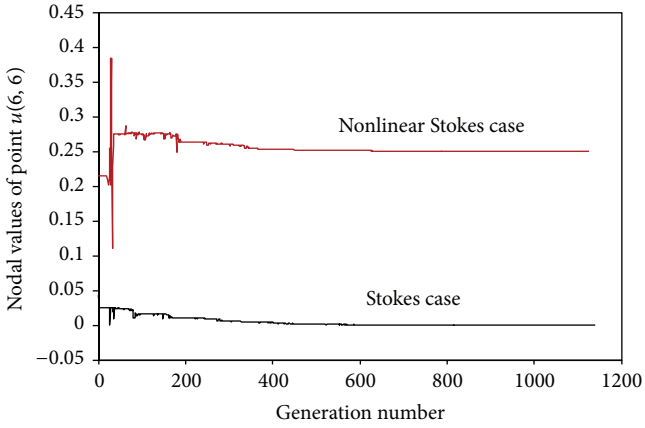

Figure 5: Nodal values of the point $u(6, 6)$ along all generations for Stokes and nonlinear Stokes cases.

Maximum fitness was reached with smooth increasing toward the maximum value along all generations and this assures how much the solution of CGA is close to the true solution. This behavior is shown in Figure 7(a).

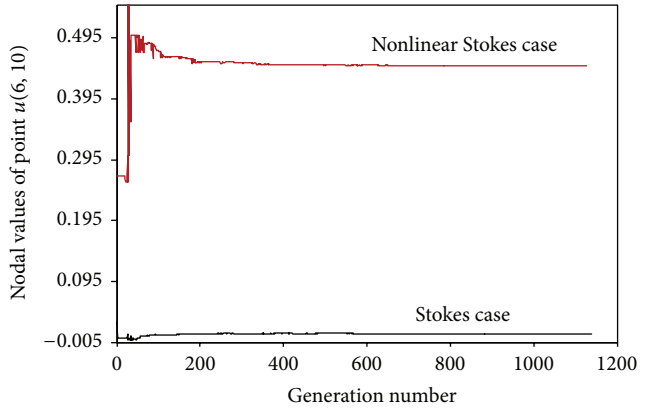

Figure 6: Nodal values of the point $u(6, 10)$ along all generations for Stokes and nonlinear Stokes cases.

4. Numerical Results for Nonlinear Navier Stokes Equations

The following conditions were taken into consideration while solving the problem.

Boundary conditions are as follows:

$$\begin{aligned} u(0, y) &= 0, & u(x, 0) &= 0, \\ u(1, y) &= y, & u(x, 1) &= x, \\ v(0, y) &= 0, & v(x, 0) &= 0, \\ v(1, y) &= y^2, & v(x, 1) &= x^2. \end{aligned} \tag{26}$$

The exact solutions are as follows:

$$\begin{aligned} u &= xy, \\ v &= x^2 y^2. \end{aligned} \tag{27}$$

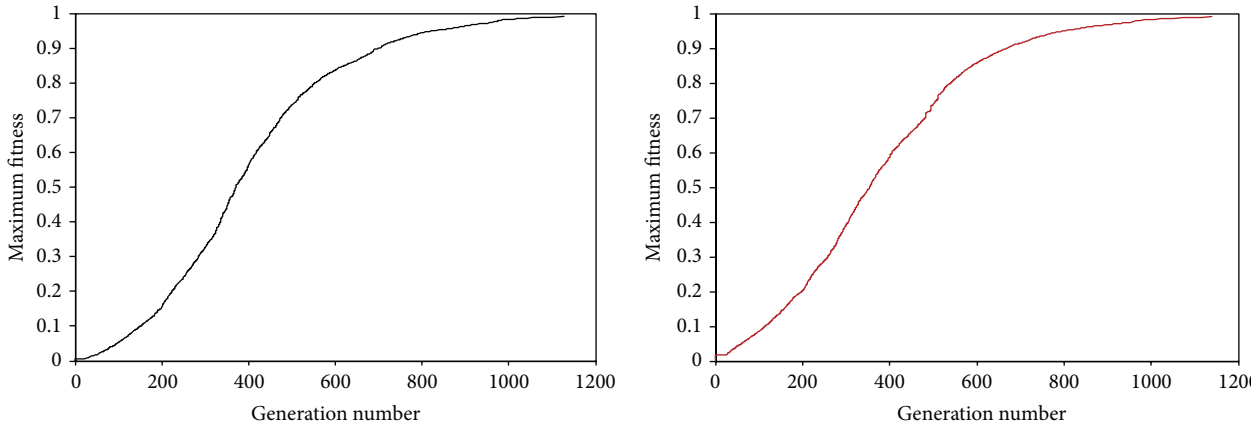

Figure 7: Maximum fitness values along all generations for Stokes and nonlinear Stokes cases.

Table 10: Actual v nodal values for nonlinear Navier Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.0001	0.0004	0.0009	0.0016	0.0025	0.0036	0.0049	0.0064	0.0081	0.01
y_2	0	0.0004	0.0016	0.0036	0.0064	0.01	0.0144	0.0196	0.0256	0.0324	0.04
y_3	0	0.0009	0.0036	0.0081	0.0144	0.0225	0.0324	0.0441	0.0576	0.0729	0.09
y_4	0	0.0016	0.0064	0.0144	0.0256	0.04	0.0576	0.0784	0.1024	0.1296	0.16
y_5	0	0.0025	0.01	0.0225	0.04	0.0625	0.09	0.1225	0.16	0.2025	0.25
y_6	0	0.0036	0.0144	0.0324	0.0576	0.09	0.1296	0.1764	0.2304	0.2916	0.36
y_7	0	0.0049	0.0196	0.0441	0.0784	0.1225	0.1764	0.2401	0.3136	0.3969	0.49
y_8	0	0.0064	0.0256	0.0576	0.1024	0.16	0.2304	0.3136	0.4096	0.5184	0.64
y_9	0	0.0081	0.0324	0.0729	0.1296	0.2025	0.2916	0.3969	0.5184	0.6561	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 11: Nodal values of u using finite element method in COMSOL for the nonlinear case.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00114	0.00238	0.00389	0.0059	0.00881	0.01335	0.0209	0.03432	0.06002	0.1
y_2	0	0.00214	0.00445	0.00723	0.01095	0.01641	0.02519	0.04074	0.07091	0.12767	0.2
y_3	0	0.003	0.00621	0.01008	0.01535	0.02337	0.03714	0.06365	0.11641	0.20307	0.3
y_4	0	0.00379	0.00781	0.01272	0.01956	0.03052	0.05059	0.09103	0.16659	0.27348	0.4
y_5	0	0.00461	0.00947	0.01544	0.02399	0.03828	0.06577	0.12076	0.21378	0.33949	0.5
y_6	0	0.00568	0.01155	0.01872	0.02913	0.04717	0.08281	0.15151	0.25716	0.40327	0.6
y_7	0	0.00757	0.01506	0.02385	0.03642	0.05846	0.10225	0.18249	0.29951	0.46673	0.7
y_8	0	0.01211	0.02323	0.03537	0.05133	0.07788	0.12894	0.21691	0.34411	0.53192	0.8
y_9	0	0.02659	0.04985	0.07414	0.10151	0.13769	0.19393	0.2804	0.41185	0.61123	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

The second terms in the right hand side of (16) and (17) are defined in (28) and they were obtained by substituting the exact solutions given in (27) into (16) and (17):

$$f_x = x^3 y^2 + xy^2,$$
$$f_y = 2x^4 y^3 + 2x^2 y^3 - 2x^2 - 2y^2. \quad (28)$$

The actual nodal values of both u and v variables are shown in Tables 9 and 10. The graphical representations of those actual values are shown in Figures 8(a) and 8(b).

Tables 11 and 12 show the obtained nodal values of u and v variables as a result of applying COMSOL. Moreover, Figures 8(c) and 8(d) show the graphical representations of these solution values.

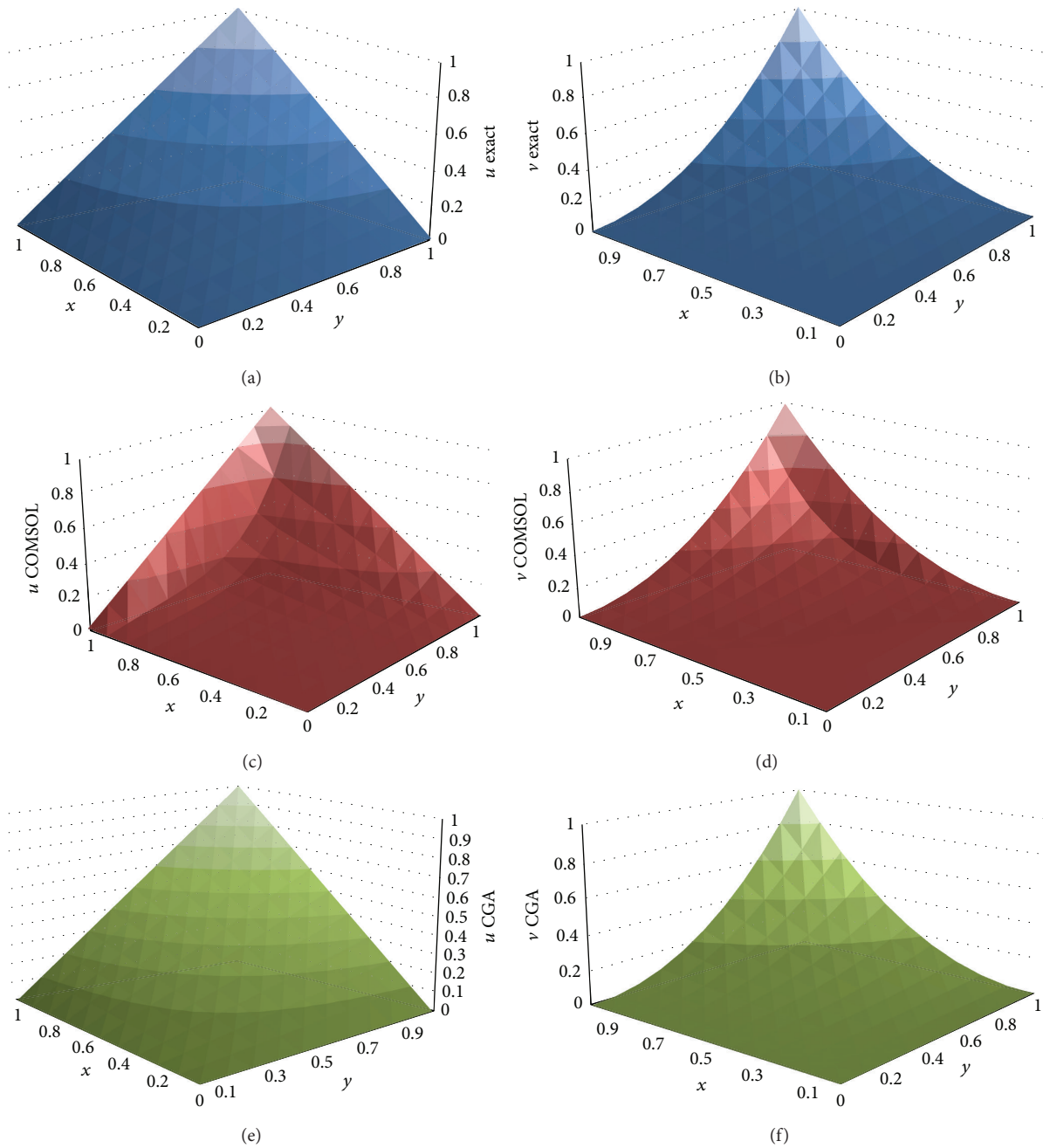

FIGURE 8: For nonlinear Navier Stokes. (a) Graphical representation for actual values of u. (b) Graphical representation for actual values of v. (c) Finite element solution for u using COMSOL. (d) Finite element solution for v using COMSOL. (e) CGA solution for u. (f) CGA solution for v.

For solution obtained by COMSOL, the norm error was found to be about 1.6258 for u variable values and about 0.5845 for v variable. And the time needed to solve this problem was about 0.546 second. For the nonlinear Navier Stokes, the results of using COMSOL to solve u variable for different sizes of meshes are shown in Table 13.

4.1. Solution of Nonlinear Navier Stokes Using Advanced Continuous Genetic Algorithm. Firstly, the discretization of (16) is

$$u_{i,j}\left[\frac{u_{i+1,j}-u_{i-1,j}}{2h_x}\right]+v_{i,j}\left[\frac{u_{i,j+1}-u_{i,j-1}}{2h_y}\right]=\left[\frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h_x^2}+\frac{u_{i,j-1}-2u_{i,j}+u_{i,j-1}}{h_y^2}\right]+f_x \tag{29}$$

Table 12: Nodal values of v using finite element method in COMSOL for nonlinear case.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00117	0.00205	0.00265	0.0031	0.00333	0.0033257	0.0027	0.00068	0.00413	0.01
y_2	0	0.00218	0.00351	0.00410	0.0042	0.00398	0.0032	0.0012	0.00363	0.01451	0.04
y_3	0	0.00302	0.00435	0.00441	0.0036	0.00229	4×10^{-6}	0.00404	0.0118	0.02991	0.09
y_4	0	0.00362	0.0045	0.00349	0.0013	0.0021	0.0066	0.01319	0.02459	0.05177	0.16
y_5	0	0.00385	0.00375	0.00101	0.0036	0.0096	0.01758	0.02864	0.04622	0.08582	0.25
y_6	0	0.00355	0.00171	0.00374	0.0116	0.0222	0.0368	0.05721	0.08729	0.14279	0.36
y_7	0	0.00243	0.00232	0.01169	0.0251	0.0446	0.07346	0.11277	0.16266	0.23196	0.49
y_8	0	6.4×10^{-5}	0.00922	0.02522	0.0495	0.0878	0.14193	0.20487	0.27293	0.35907	0.64
y_9	0	0.00398	0.02085	0.049881	0.0956	0.163	0.24296	0.32916	0.42668	0.54521	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 13: COMSOL results for different meshes.

Number of elements in mesh	L_2 norm error	CPU time (s)
100	1.6258	0.55
228	0.0213	0.57
365	0.0207	0.702
798	0.0201	1.014
2736	0.0195	2.386
24865	0.019	18.034

and is the same for the second equation. Then, for residues, one has

$$r_{1i,j} = u_{i,j}\left[\frac{u_{i+1,j} - u_{i-1,j}}{2h_x}\right] + v_{i,j}\left[\frac{u_{i,j+1} - u_{i,j-1}}{2h_y}\right] - \left[\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}\right] - f_x, \tag{30}$$

Overall residues are the summation for both first and second equations residues:

$$R1 = \sqrt{\sum_{i=1}^{n_x}\sum_{j=1}^{m_y} r_{1i,j}^2}, \qquad R2 = \sqrt{\sum_{i=1}^{n_x}\sum_{j=1}^{m_y} r_{2i,j}^2}, \tag{31}$$

$$R = R1 + R2.$$

A solution using continuous genetic algorithm (CGA) for case 2 (nonlinear Navier Stokes equation [Dirichlet boundary conditions]) is shown in Tables 14 and 15, including the nodal values of the boundaries. And the graphical representation of these values is shown in Figures 8(e) and 8(f).

The results of applying CGA can be summarized in calculating the L_2 norm error that was found to be about 1.42×10^{-4} for u variable values and about 9.76×10^{-5} for v variable values. The time that CGA needed to solve such kind of problems was about 7456 seconds.

It can be concluded from the results that solving nonlinear equations using finite element method in COMSOL application have a low accuracy. On the other hand, the trend of accuracy for CGA method is the same for both linear and nonlinear cases.

The CGAs process for solving nonlinear Navier Stokes equations in visual basic tools about 7456 seconds, in which the average number of generations was 1127 generations. Moreover, the average residual value was about 5.75433×10^{-4}. Then, the average fitness values according to the equation $F = 1/(1 + R) = 0.9994$ are very close to the maximum fitness value of unity.

The convergence state of nodal values of the nodes was presented along all generations such as an example for [$u(2, 2)$, $u(6, 6)$, and $u(6, 10)$] nodes excluding the boundaries values as shown in Figures 4, 5, and 6, respectively. It is concluded from the figures that steady state is reached for $u(2, 2)$ nodes much earlier than that for $u(6, 6)$ and $u(6, 10)$. The reason for that is that the node $(2, 2)$ location is much closer to the boundaries compared with the other nodes locations. For the same reason, the convergence for node $(6, 10)$ is faster than for $(6, 6)$.

The nodal residues in this case were less than 22.1059×10^{-4} for u values as shown in Table 16 and were less than 25.686×10^{-4} for v values as shown in Table 17. Maximum fitness was reached with smooth increasing toward the maximum value along all generations and this assures how much the solution of CGA is close to the true solution. This behavior is shown in Figure 7(b).

Table 18 summarizes the overall performance of the used solvers solutions for the selected cases of Navier Stokes equations. As shown in the table, the CGA method has good results in solving both linear and nonlinear problems. It is worth mentioning that in the Stokes case the accuracy of COMSOL was higher than that of CGA and Wang et al. results. However, in the nonlinear case, COMSOL results' errors were higher than those of the CGA results.

As a result, it can be concluded that COMSOL software has good predictions and small errors for the linear cases

Table 14: Nodal u values including boundaries for nonlinear Navier Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.01002	0.020019	0.03002	0.04002	0.05001	0.06	0.07	0.08	0.09	0.1
y_2	0	0.02002	0.040022	0.06003	0.08002	0.10001	0.12001	0.14001	0.16	0.18	0.2
y_3	0	0.03002	0.06002	0.09003	0.12002	0.15002	0.18001	0.21001	0.24001	0.27	0.3
y_4	0	0.04	0.080009	0.12002	0.16002	0.20002	0.24002	0.28002	0.32002	0.36001	0.4
y_5	0	0.04999	0.1	0.15001	0.20002	0.25002	0.30002	0.35003	0.40002	0.45001	0.5
y_6	0	0.05998	0.119997	0.18	0.24002	0.30002	0.36002	0.42003	0.48003	0.54002	0.6
y_7	0	0.06999	0.139998	0.21001	0.28002	0.35002	0.42002	0.49003	0.56003	0.63002	0.7
y_8	0	0.07999	0.160001	0.24001	0.32001	0.40001	0.48002	0.56002	0.64002	0.72001	0.8
y_9	0	0.09001	0.180006	0.27001	0.36001	0.45001	0.54001	0.63001	0.72001	0.81001	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

Table 15: Nodal v values including boundaries for nonlinear Navier Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00011	0.0004	0.0009	0.0016	0.00251	0.0036	0.0049	0.0064	0.0081	0.01
y_2	0	0.00039	0.0016	0.0036	0.00641	0.01001	0.0144	0.0196	0.02561	0.0324	0.04
y_3	0	0.00088	0.00359	0.00809	0.01441	0.02251	0.0324	0.0441	0.05761	0.0729	0.09
y_4	0	0.00158	0.00638	0.01439	0.0256	0.04001	0.0576	0.0784	0.10241	0.1296	0.16
y_5	0	0.00249	0.00999	0.02249	0.04	0.06251	0.09	0.1225	0.16001	0.2025	0.25
y_6	0	0.0036	0.0144	0.0324	0.0576	0.09	0.1296	0.1764	0.23041	0.2916	0.36
y_7	0	0.00492	0.01961	0.04411	0.07839	0.12249	0.1764	0.2401	0.3136	0.3969	0.49
y_8	0	0.00642	0.02561	0.05761	0.10239	0.15999	0.2304	0.3136	0.4096	0.51841	0.64
y_9	0	0.00813	0.03242	0.07291	0.12959	0.20248	0.2916	0.3969	0.5184	0.65611	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 16: Residual u values for all nodes for nonlinear Navier Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	3.2678	3.4335	2.66768	5.61671	2.83248	0.4064	3.16061	1.10452	2.13471	0
y_2	0	8.4514	1.4042	4.34961	1.19837	0.67391	2.3282	6.07949	4.44807	4.88446	0
y_3	0	4.3892	3.1142	2.07415	11.8851	1.81051	1.3184	22.1059	11.5826	5.51202	0
y_4	0	10.995	3.8016	13.5119	14.0723	0.54436	2.6779	9.73838	8.83235	0.27101	0
y_5	0	11.113	1.4708	6.96716	11.347	6.3083	4.7194	9.58696	7.05931	2.25748	0
y_6	0	7.9889	1.4367	0.50162	5.15927	1.85226	2.642	11.4612	3.56919	0.78898	0
y_7	0	8.3608	3.1107	30.6914	6.85968	3.82915	0.8626	0.45646	5.65133	0.51875	0
y_8	0	2.1557	6.875	18.7506	4.19258	0.02673	2.7288	5.30467	0.00834	3.6369	0
y_9	0	2.1154	4.5035	4.44288	5.03227	0.89008	2.042	0.79609	1.67729	0.46048	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

but not for nonlinear problems since its accuracy dramatically deceased. This makes CGA technique advantageous over the COMSOL technique because of its adaptability to the nonlinear nature of the problems with reasonable and consistent errors. However, it should be mentioned that the CGA method requires longer time periods than COMSOL in solving the problems. This is a new challenge for CGA, which could be overcome through the continuous development of the CGA algorithm or the computer hardware. However, it is worth mentioning that CGA has good accuracy even in the first run of solving such kind of problems.

5. Conclusions

In this paper, two-dimensional CGA was applied for the solution of steady-state two-dimensional Stokes and nonlinear Navier Stokes problems. The proposed CGA for the solution

Table 17: Residual v values for all nodes for nonlinear Navier Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	6.5086	4.0087	12.004	11.414	11.025	18.179	1.3779	0.3946	4.3496	0
y_2	0	0.2215	0.2655	2.231	2.344	2.9315	5.6304	7.1332	6.5507	1.8278	0
y_3	0	16.501	0.157	21.686	19.681	10.5	11.835	0.8445	3.0172	6.0424	0
y_4	0	6.6381	4.0749	0.8183	4.7228	8.8724	21.16	7.4542	8.3205	3.6702	0
y_5	0	2.9267	8.0611	13.552	2.0773	0.2907	10.927	5.7256	3.4446	1.6143	0
y_6	0	18.818	8.7243	25.686	3.0502	4.1499	3.4448	1.7468	2.8593	4.1917	0
y_7	0	5.8223	2.5064	11.64	9.513	7.5191	14.893	2.2233	2.9787	7.9392	0
y_8	0	12.419	6.9534	2.7893	10.201	13.735	1.7175	6.0922	4.7986	2.2204	0
y_9	0	2.2476	1.6574	3.4965	2.8279	3.9258	1.4725	0.6767	4.4832	0.8819	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 18: Summary of all Navier Stokes cases results.

	Solution method	Average L_2 norm error (according to exact solution)	CPU time (seconds)	Tolerance and termination criteria
(Stokes case)	CGA	2.05×10^{-4}	7523	Fitness = 0.99 or number of generations = 3000
	Finite element Wang et al., (2009) [24]	5.05×10^{-4}	—	1×10^{-8}
	Galerkin finite element in COMSOL	1.25×10^{-5}	0.045	1×10^{-6}
(Nonlinear Navier Stokes case)	CGA	1.198×10^{-4}	7456	Fitness = 0.99 or number of generations = 3000
	Galerkin finite element in COMSOL	1.1052	0.546	1×10^{-6}

of partial differential equations has the following advantages: first, it does not require any modification while switching from the linear to the nonlinear case, only the discretization of the problem will be changed, while the solution procedure using CGA will be the same; second, the algorithm requires minimal knowledge about the aimed problem that is the discretized form of the partial differential equation in the residual form; third, the method is found to be simple, efficient, and attractive with great potential in engineering applications. The resulting solutions provided good accuracy for Stokes case and excellent accuracy as compared to other commercial packages for the nonlinear case.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. B. Abbott and D. R. Basco, *Computational Fluid Dynamics: An Introduction for Engineers*, Long Man Group, New York, NY, USA, 1989.

[2] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, Springer, New York, NY, USA, 1996.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[4] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1996.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[6] L. Davis, *Hand Book of Genetic Algorithms*, Van Norstrand Reinhold, New York, NY, USA, 1991.

[7] Z. S. Abo-Hammour, *Advanced continuous genetic algorithms and their applications in the motion planning of robot manipulators and in numerical solution of boundary value problems [Ph.D. thesis]*, Quaid-i-Azam University, Islamabad, Pakistan, 2002.

[8] Z. S. Abo-Hammour, M. Yusuf, N. M. Mirza, S. M. Mirza, M. Arif, and J. Khurshid, "Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 8, pp. 1219–1242, 2004.

[9] H. Jaradat, *Numerical solution of temporal two point boundary value problems using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2006.

[10] O. Abu Arqub, Z. Abo-Hammour, S. Momani, and N. Shawagfeh, "Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic

algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 401696, 15 pages, 2014.

[11] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems," *Applied Mathematics and Information Systems*, vol. 8, no. 1, Article ID 205391, pp. 235–248, 2014.

[12] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "An optimization algorithm for solving systems of singular boundary value problems," *Applied Mathematics and Information Systems*, 2014.

[13] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "Solving singular two-point boundary value problems using continuous genetic algorithm," *Abstract and Applied Analysis*, vol. 2012, Article ID 205391, 25 pages, 2012.

[14] Z. S. Abo-Hammour, R. B. Albadarneh, and M. S. Saraireh, "Solution of Laplace equation using continuous genetic algorithms," *Kuwait Journal of Science and Engineering A*, vol. 37, no. 2, pp. 1–15, 2010.

[15] O. F. Al-Sayyed, *Numerical solution of differential-algebraic equations using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2006.

[16] O. M. Abo-Arqoub, *Numerical solution of fuzzy differential equations using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2008.

[17] A. Samhouri, *Applicability of continuous genetic algorithm in solving navier stokes problems [M.S. thesis]*, University of Jordan, Amman, Jordan, 2012.

[18] Z. S. Abo-Hammour, N. M. Mirza, S. M. Mirza, and M. Arif, "Cartesian path generation of robot manipulators using continuous genetic algorithms," *Robotics and Autonomous Systems*, vol. 41, no. 4, pp. 179–223, 2002.

[19] Z. S. Abo-Hammour, O. M. Alsmadi, S. I. Bataineh, M. A. Al-Omari, and N. Affach, "Continuous genetic algorithms for collision-free Cartesian path planning of robot manipulators regular paper," *International Journal of Advanced Robotic Systems*, vol. 8, no. 6, pp. 14–36, 2011.

[20] O. M. K. Alsmadi, Z. S. Abo-Hammour, A. M. Al-Smadi, and M. S. Saraireh, "A robust and efficient genetic algorithm for solving a chemical reactor problem: theory, application and convergence analysis," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 5, pp. 594–603, 2012.

[21] Z. S. Abo-Hammour, A. G. Asasfeh, A. M. Al-Smadi, and O. M. K. Alsmadi, "A novel continuous genetic algorithm for the solution of optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 414–432, 2011.

[22] L. L. Lapin, *Probability and Statistics for Modern Engineering*, PWS-KENT, Boston, Mass, USA, 2nd edition, 1990.

[23] S. C. R. Dennis and J. D. Hudson, "Compact h4 finite-difference approximations to Operators of Navier-Stokes Type," *Journal of Computational Physics*, vol. 85, no. 2, pp. 390–416, 1989.

[24] J. Wang, Y. Wang, and X. Ye, "A robust numerical method for stokes equations based on divergence-free iidie(div) finite element methods," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2784–2802, 2009.