ITALIAN JOURNAL OF PURE AND APPLIED MATHEMATICS – N. 45–2021 (216–236)

A novel continuous genetic algorithm technique for the solution of partial differential equations

Ramzi B. Albadarneh*
Department of Mathematics
Faculty of Science
The Hashemite University
Zarqa 13115
Jordan
rbadarneh@hu.edu.jo

Zaer Abo-Hammour
Department of Mechatronics Engineering
The University of Jordan
Amman 11942
Jordan
zaer@ju.edu.jo

Othman Alsmadi
Department of Electrical Engineering
The University of Jordan
Amman 11942
Jordan
othmanmk@ju.edu.jo

Nabil Shawagfeh
Department of Mathematics
The University of Jordan
Amman 11942
Jordan
shawagnt@ju.edu.jo

Abstract. In this paper, a novel continuous genetic algorithm (CGA) approach is proposed for the solution of Laplace, Poisson, Helmholtz, and nonlinear partial differential equations (PDEs) due to their importance as they are encountered in a variety of mathematical and physical systems. The approach is formulated by firstly converting the equation into an algebraic equation using the finite difference scheme which approximates its partial derivatives. Then, the residual value of each interior node is calculated based on the nodal values generated by the CGA. The solution is finally obtained based on the minimization of the overall residual values and correspondingly maximizing the fitness value. Different test problems were considered in this work, which cover different boundary conditions including Dirichlet and Neumann boundary conditions. The solution obtained using the proposed CGA approach is compared with the exact solutions

---

*. Corresponding author

along with the finite difference and finite element methods. Numerical results show the strength, potentiality, and superiority of the proposed CGA approach.

Keywords: numerical methods, continuous genetic algorithms, partial differential equations, finite elements method, finite difference method.

1. Introduction

The exact solutions of PDEs are usually unavailable as reported by Morton et al. and others [28, 41]. Therefore, numerical approximated solutions of these equations are usually considered. In the modeling of natural phenomena, a crucial role is played by the study of PDEs of elliptic type as they arise in every field of science. Consequently, the desire to obtain accurate solutions of these equations has always a prominent place in the efforts of different scientists [44]. Due to their importance in science and engineering, PDEs are studied extensively by specialists and practitioners [48]. Numerical differentiation, on the other hand, plays an important role in solving ordinary differential equations (ODEs) and PDEs, which have been developed to derive useful difference formulas for numerical derivatives. Most techniques of numerical differentiation are classified into the following types [31]: finite difference type[16, 17, 34, 43], polynomial interpolation type [16, 17, 19, 20, 22, 30, 36, 38], [43, 45], operator type [47], lozenge diagrams [21], and undetermined coefficients [21, 8]. For a brief and interesting survey, the interested reader may consult [31]. In this paper, we will consider the solution of Laplace, Poisson, Helmholtz, and nonlinear PDEs. The Laplace and Poisson's equations have been considered by many scientists as it arises in a variety of mathematical and physical systems, ranging through fluid mechanics, heat conduction, electromagnetism, potential theory, solid mechanics, geometry, probability, number theory, etc. [27, 29, 33, 39, 49]. In the study of physical problems involving PDEs in both space and time, Helmholtz equation plays an important role as well. The equation arises when one is looking for mono-frequency or time-harmonic solutions to the wave equation [18]. Because of its relationship to the wave equation, the Helmholtz equation becomes important in the study of electromagnetic radiation, seismology, and acoustics [26, 50]. Despite the apparent simplicity of the underlying differential relations, nonlinear PDEs govern a great deal of complex phenomena of motion, reaction, diffusion, equilibrium, conservation, and more. In recent years, a variety of effective methods have been proposed for the solution of nonlinear PDEs as seen in the reaction-diffusion equations [51], lattice differential equations [15], and the heat equation [52]. Genetic algorithms (GAs) have also been applied to solve problems in computer science, computational physics, molecular chemistry, statistics and applied probability. Recently, they were applied for model order reduction [14], ARMA modeling [5], and system identification [4]. In 2002, Abo-Hammour developed continuous genetic algorithms (CGAs) as an efficient method for the solution of optimization problems [8]. This development has opened the doors for wide applications of the algorithm in the field of engineering and mathematics. The CGAs have been successfully applied in the solution of optimal control problems [6], collision-free Cartesian planning of robot manipulators [7], singular two-point boundary value problems [3], higher-order initial value problems [2], and nonlinear boundary value problems [1].

2. Problem formulation

A first-order partial differential equation with n independent variables has the general form

$$F(x_1, x_2, \cdots, x_n,\ u_{x_1}, u_{x_2}, \cdots, u_{x_n}) = 0, \tag{1}$$

where $u(x_1, x_2, \cdots, x_n)$ is the unknown function and $F\ (...)$ is a given function.

A second-order nonlinear partial differential equation with two independent variables has the form

$$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0. \tag{2}$$

The classification of Equation (2) depends on the discriminate $\sigma = b^2 - ac$, where

$$a = \frac{\partial F}{\partial p}, b = \frac{1}{2}\frac{\partial F}{\partial q}, c = \frac{\partial F}{\partial r}, p = u_{xx}, q = u_{xy}, \text{ and } r = u_{yy}.$$

Thus, the classification is made such that $\sigma > 0$ is hyperbolic, $\sigma = 0$ is parabolic, and $\sigma < 0$ is elliptic [42].

In this paper, we will consider the linear PDEs, which involve Laplace equation, Poisson equation, and Helmholtz equation, and different types of nonlinear equations given as following:

Linear equations:

Laplace equation:

$$\nabla^2 u(x, y) = 0. \tag{3}$$

Poisson equation:

$$\nabla^2 u(x, y) = h(x, y). \tag{4}$$

Helmholtz equation:

$$\nabla^2 u(x, y) + g(x, y)u(x, y) = h(x, y), \tag{5}$$

For a general Nonlinear form, the equation may look like

$$(u^{n_0} + a_0)\nabla^2 u = a_1(x, y)u^{n_1} + a_2(x, y)u^{n_2} + a_3(x, y)u^{n_3} + a_4(x, y), \tag{6}$$

where a_0, n_i are real number for $i = 0, 1, 2, 3$, $a_i(x, y)$ are functions of x and y for $i = 1, 2, 3, 4$.

Since the exact solution for many partial differential equations usually does not exist, numerical methods for solving such equations are suggested. The most widely known numerical methods used for such purpose are the finite difference methods (FDMs) [9, 10] and finite element methods (FEMs).

The proposed genetic algorithm approach, for numerically approximating the solution, requires a mapping of the individual residual, R, into a fitness function, F, in order to convert the minimization problem of R into a maximization problem of F. For the Laplace equation, in order to find the fitness function, the finite difference formulas are applied to Equation (3) as following

$$\frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h^2}+\frac{u_{i,j-1}-2u_{i,j}+u_{i,j+1}}{k^2}=0, \tag{7}$$

where $u_{i,j} = u(x_i,\ y_j)$, h is the step size of the nodes x and k is the step size of the nodes y. Similarly, for the Poisson equation, the following is obtained

$$\frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h^2}+\frac{u_{i,j-1}-2u_{i,j}+u_{i,j+1}}{k^2}=h_{i,j}, \tag{8}$$

where $h_{i,j} = h(x_i,\ y_j)$. Also, for the Helmholtz equation, the following is obtained

$$\frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h^2}+\frac{u_{i,j-1}-2u_{i,j}+u_{i,j+1}}{k^2}+g_{i,j}u_{i,j}=h_{i,j}, \tag{9}$$

where $g_{i,j} = g(x_i,\ y_j)$. For the general nonlinear form, the fitness function can be obtained based on the observation that Equation (6) implies that

$$u_{i,j}=f(x_i,y_i,u_{i-1,j},u_{i+1,j},u_{i,j-1},u_{i,j+1}), \tag{10}$$

where

$$\begin{aligned}&f(x_i,y_i,u_{i-1,j},u_{i+1,j},u_{i,j-1},u_{i,j+1})\\&=\left(\frac{h^2k^2}{2h^2+2k^2-h^2k^2g_{i,j}}\right)\left(\frac{u_{i-1,j}+u_{i+1,j}}{h^2}+\frac{u_{i-1,j}+u_{i+1,j}}{k^2}-h_{i,j}\right).\end{aligned} \tag{11}$$

Then, the residual at each interior node is

$$r_{i,j}=u_{i,j}-f(x_i,y_i,u_{i-1,j},u_{i+1,j},u_{i,j-1},u_{i,j+1}). \tag{12}$$

Hence, the residual of the nonlinear equation in (6) is given by

$$\begin{aligned}&r_{i,j}=u_{i,j}\left(\frac{2}{h^2}+\frac{2}{k^2}\right)(u_{i,j}^{n_0}+a_0)+u_{i,j}^2a_1(i,j)u_{i,j}^{n_1}+a_2(i,j)u_{i,j}^{n_2}+\\&a_3(i,j)u_{i,j}^{n_3}+a_4(i,j)-(u_{i,j}^{n_0}+a_0)\left(\frac{u_{i-1,j}+u_{i+1,j}}{h^2}+\frac{u_{i-1,j}+u_{i+1,j}}{k^2}\right).\end{aligned} \tag{13}$$

As a result, the overall residual at each parent can be written as

$$R = \left(\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} r_{i,j}^2 \right)^{0.5} . \tag{14}$$

Thus, the suitable fitness function can then be given as

$$F = \frac{1}{1+R}. \tag{15}$$

3. Continuous genetic algorithm (CGA)

The main operators of CGA [23] have smooth characteristics and avoid sharp jumps in the parameter values. First, the random initialization using continuous surfaces reduce the possibility of highly oscillating values in the neighboring parameters. Second, the crossover operator(s) using continuous surfaces result in smooth transition in the parameter values. Third, the mutation process changes the value of the parameter in smooth transition in the parameter values. In summary, traditional GA type results in a step-function-like jump in the parameter values while CGA results in smooth transitions, for more details see [8].

The advantages of CGA over conventional GA when it is applied to problems with many parameters [8] are seen as following:

1. In traditional GA, the actual range of the nodal values should lie within the expected range during the evolution process; otherwise we will not be able to reach to the optimal solution. This is because the encoding/decoding processes in conventional GA require the range of values within which the solution should lie. To overcome this problem, we provide a wide range for the solution values. In CGA, we initialize the population to lie within some initial guesses and the GA will use the overall residual values throughout the evolution process.

2. There are no encoding/decoding processes in CGA. This means that the execution time of an optimization problem will be smaller in this case if both GA versions converge in approximately the same number of generations.

3. The memory requirements of traditional GA are higher than that of the CGA because the former uses genotype representation of the population individuals while the later utilizes only the phenotype data. This makes CGA more appropriate to problems of larger number of parameters.

The continuous genetic algorithm proposed in this work (which is generalization of [8]) consists of the following steps:

1. Initialization: The initial population is divided into four equal segments; each of them is generated using a different set of smooth functions to generate various initial populations. The first part of the initial population is generated using linear combination of one-dimensional Gaussian functions with respect to x and of one-dimensional Gaussian functions with respect to y, the second segment of the initial population, is generated using linear combination of one-dimensional tangent hyperbolic functions with respect to x and one-dimensional tangent hyperbolic functions with respect to y, the third part is generated using cosine function while the fourth one is sine function. The four functions are given as following

$$\begin{aligned}
&g_1(x_i,y_j)=\frac{i}{i+j}\left[a_1(x_i)+(b_1(x_i)-a_1(x_i))\frac{j-1}{M_y}+c\exp\left(\frac{-(j-\mu_x)^2}{2\sigma_x^2}\right)\right]\\
&+\frac{j}{i+j}\left[a_2(y_j)+(b_2(y_j)-a_2(y_j))\frac{j-1}{M_x}+c\exp\left(\frac{-(i-\mu_y)^2}{2\sigma_y^2}\right)\right],
\end{aligned}\tag{16}$$

$$\begin{aligned}
&g_2(x_i,y_j)=\frac{i}{i+j}\left[a_1(x_i)+0.5(b_1(x_i)-a_1(x_i))\left(1+\tanh\left(\frac{j^d-\mu_x}{\sigma_x}\right)\right)\right]\\
&+\frac{j}{i+j}\left[a_2(y_j)+0.5(b_2(y_j)-a_2(y_j))\left(1+\tanh\left(\frac{i^d-\mu_y}{\sigma_y}\right)\right)\right],
\end{aligned}\tag{17}$$

$$\begin{aligned}
&g_3(x_i,y_j)=a_1(x_i)+(b_1(x_i)-a_1(x_i))\\
&\cdot\left[1-\cos\left(\pi d_1+\frac{\pi}{4}\left(c_1\tanh\left(\frac{2d\times j-\mu_x}{\sigma_x}\right)+1\right)\right)\right]\\
&+b_2(y_j)-(b_2(y_j)-a_2(y_j))\\
&\cdot\left[1-\cos\left(\pi d_1+\frac{\pi}{4}\left(c_1\tanh\left(\frac{2d\times i-\mu_y}{\sigma_y}\right)+1\right)\right)\right],
\end{aligned}\tag{18}$$

$$\begin{aligned}
&g_4(x_i,y_j)=a_1(x_i)+(b_1(x_i)-a_1(x_i))\\
&\sin\left(\pi d\left(\tanh\left(\frac{2d_2\times j-\mu_x}{\sigma_x}\right)+1\right)\right)\\
&+a_2(y_j)-(b_2(y_j)-a_2(y_j))\sin\left(\pi d\left(\tanh\left(\frac{2d_2\times i-\mu_y}{\sigma_y}\right)+1\right)\right),
\end{aligned}\tag{19}$$

where $1\le i\le M_{x+1}$, $1\le j\le M_{y+1}$, M_x, M_y are the number of nodes of x and y respectively, c is a random number in the range [-0.5, 1], d is a random number in the range [0.3, 0.8], d_1 is a random number in the range [0.05, 0.06], d_2 is a random number in the range [0, 1], c_1 is a random number in the range [0.3, 1.8], $a_1(x_i)$, $a_2(y_j)$, $b_1(x_i)$, $b_2(y_j)$ represent the boundary values, μ_x is a random number in the range [$M_x/4$, $M_y/4$], σ_x is random number in the range [1, $M_x/3$], μ_y is a random number in the range [$M_y/4$, $3M_y/4$], σ_y is random number in the range [1, $M_y/3$]. It is to be noted that μ_x and μ_y specify the center of the function while σ_x and $\sigma_y$specify the degree of dispersion of the curve.

The upper function, $g(x,\ y)$, is then rescaled within the range [$Pmin$, $Pmax$], where $Pmin$, $Pmax$ represents the expected minimum and maximum initial nodal values.

The above functions are not suitable if all boundary values are equal, instead of that functions we can use suitable linear combination of $g_6(x,\ y)$ and $g_7(x,\ y)$ or we can use suitable linear combination of $g_8(x,y)$ and $g_9(x,y)$ where $g_6(x,y),...,g_9(x,y)$ are

$$g_6(x_i,y_j)=a+d\left[1-\cos\left(c\pi+\frac{cc\pi(j-1)}{M_y+1}\right)\right], \tag{20}$$

$$g_7(x_i,y_j)=a+d\left[1-\cos\left(c\pi+\frac{cc\pi(i-1)}{M_x+1}\right)\right], \tag{21}$$

$$g_8(x_i,y_j)=a+d\sin\left(\frac{cc\pi(j-1)}{M_y+1}\right), \tag{22}$$

$$g_9(x_i,y_j)=a+d\sin\left(\frac{cc\pi(i-1)}{M_x+1}\right), \tag{23}$$

where $1\leq i\leq M_x+1$, $1\leq j\leq M_y+1$, M_x, M_y are the number of nodes of x and y respectively a is the boundary value, d is a random number in the range [-2, 2], c and cc are random numbers in the range [1.9, 2.1]. Note that we can rotate the above functions to reproduce new functions which give us more variety of functions. For illustration, Figure 1 shows the initialization at some given boundary conditions.

2. Evaluation: The fitness is calculated for each individual in the population.

3. Selection: We select Individuals from the current population according to their relative fitness, in order to receive more suitable copies.

4. Crossover: The way through which information is shared among the population. The crossover process combines the features of two parent individuals, i.e., P_1 and P_2, to form two children individuals, say C_1 and C_2 as given by the equations:

$$C_1(x_i,y_j)=W(x_i,y_j)P_1(x_i,y_j)+(1-W(x_i,y_j))P_2(x_i,y_j), \tag{24}$$

$$C_2(x_i,y_j)=(1-W(x_i,y_j))P_1(x_i,y_j)+W(x_i,y_j)P_2(x_i,y_j), \tag{25}$$

where W represents the mixing function in the range [0, 1]. Some types of weight functions were tested in this work are given by the equations:

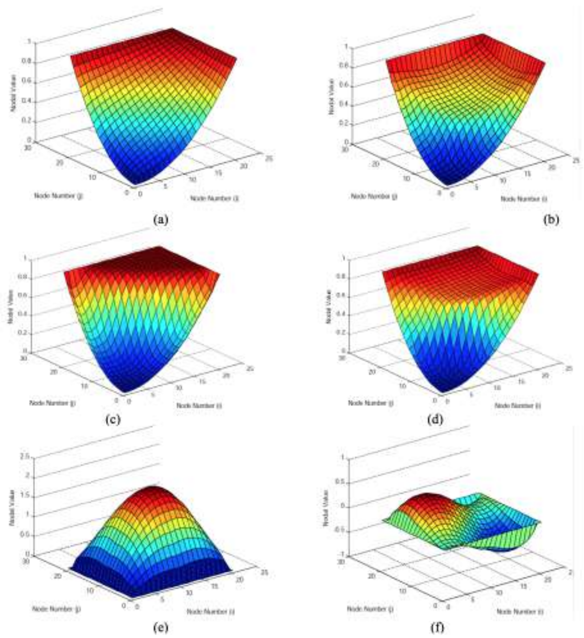


Figure 1: Two-dimensional initialization functions.

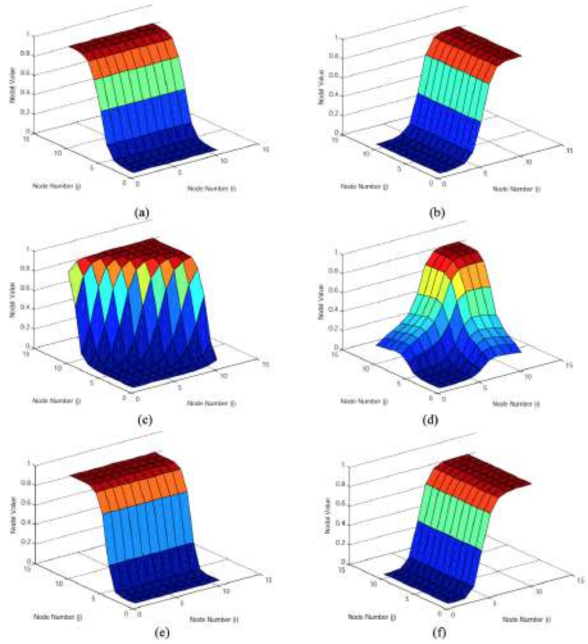


Figure 2: Two-dimensional crossover weighting functions.

$$W_1(x_i, y_j) = \frac{1}{2}\left[1 + \tanh\left(\frac{i-\mu_x}{\sigma_x}\right)\right], \tag{26}$$

$$W_2(x_i, y_j) = \frac{1}{2}\left[1 + \tanh\left(\frac{j-\mu_y}{\sigma_y}\right)\right], \tag{27}$$

$$W_3(x_i, y_j) = \frac{1}{2}\left[1 + \tanh\left(\frac{i-\mu_x}{\sigma_x} + \frac{j-\mu_y}{\sigma_y}\right)\right], \tag{28}$$

$$W_4(x_i, y_j) = (1+i)\frac{w_1(x,y)}{i+j+2} + (1+j)\frac{w_2(x,y)}{i+j+2}, \tag{29}$$

$$W_5(x_i, y_j) = 1 - \cos\left(\frac{\pi}{2}\tanh\left(\frac{i-\mu_x}{\sigma_x}\right)\right), \tag{30}$$

$$W_6(x_i, y_j) = 1 - \cos\left(\frac{\pi}{2}\tanh\left(\frac{j-\mu_y}{\sigma_y}\right)\right), \tag{31}$$

$$W_7(x_i, y_j) = 1 - \cos\left(\frac{\pi}{4}\left[\tanh\left(\frac{i-\mu_x}{\sigma_x} + \frac{j-\mu_y}{\sigma_y}\right) + 1\right]\right), \tag{32}$$

$$W_8(x_i, y_j) = (1+i)\frac{w_5(x,y)}{i+j+2} + (1+j)\frac{w_6(x,y)}{i+j+2}, \tag{33}$$

$$W_9(x_i, y_j) = \sin\left(\frac{\pi}{2}\tanh\left(\frac{i-\mu_x}{\sigma_x}\right)\right), \tag{34}$$

$$W_{10}(x_i, y_j) = \sin\left(\frac{\pi}{2}\tanh\left(\frac{j-\mu_y}{\sigma_y}\right)\right), \tag{35}$$

$$W_{11}(x_i, y_j) = \sin\left(\frac{\pi}{4}\left[\tanh\left(\frac{i-\mu_x}{\sigma_x} + \frac{j-\mu_y}{\sigma_y}\right) + 1\right]\right), \tag{36}$$

$$W_{12}(x_i, y_j) = (1+i)\frac{w_9(x_i,y_j)}{i+j+2} + (1+j)\frac{w_{10}(x_i,y_j)}{i+j+2}, \tag{37}$$

where $1 \leq i \leq M_x + 1$, $1 \leq j \leq M_y + 1$, M_x, M_y are the number of nodes of x and y respectively, σ_x and σ_y are chosen such that the tangent hyperbolic function achieves it's complete transition from -1 to 1 within the given ranges of both x and y, μ_x is a random number in the range $[M_x/3, 2M_x/3]$, μ_y is a random number in the range $[M_y/3,\ 2M_y/3]$, σ_x is random number in the range $[1,\ M_x/4]$, σ_y is random number in the range $[1,\ M_x/4]$. Figures 2 and [3] show some weighted functions and Figure 4 shows the crossover process between two random parents.

5. Mutation: The function of mutation is to make diversity within the population. The mutation process in CGA is governed by the following formulas:

$$C_m(x_i, y_j) = C(x_i, y_j) + d \times M(x_i, y_j), \tag{38}$$

where $C(x_i{,}y_j)$ represents the child produced through the crossover process, $C_m(x,\ y)$ is the mutated child and $M(x_i{,}y_j)$ is the mutation function, d represents a random number within the range $[-R_{ave}{,}R_{ave}]$, R_{ave} is average overall

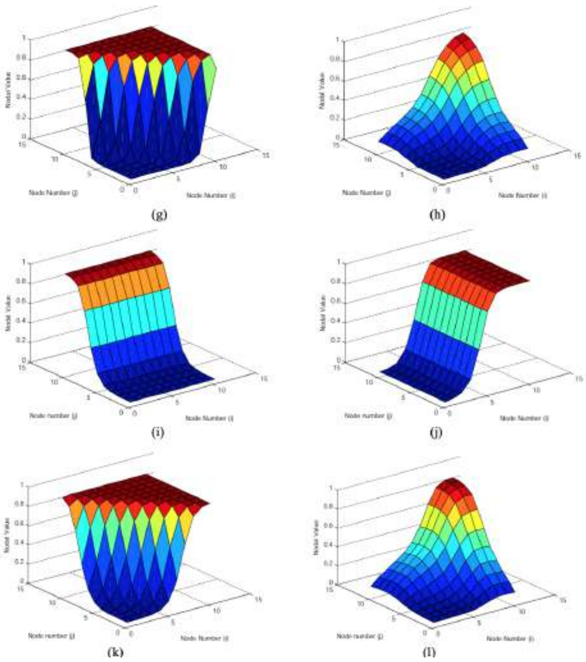


Figure 3: Two-dimensional crossover weighting functions.

residue of the population. The mutation functions are $M_1(x_i,y_j)$, linear combination of M_2 and M_3 or linear combination of M_4 and M_5 were M_1, M_2, M_3, M_4 and M_5 are

$$M_1(x_i,y_j) = \exp\left(-\frac{(i-\mu_x)^2}{2\sigma_x^2} - \frac{(j-\mu_y)^2}{2\sigma_y^2}\right), \tag{39}$$

$$M_2(x_i,y_j) = d\left(1-\cos\left(\frac{(i-1)c\pi}{N-1}\right)\right), \tag{40}$$

$$M_3(x_i,y_j) = d\left(1-\cos\left(\frac{(j-1)c\pi}{N-1}\right)\right), \tag{41}$$

$$M_4(x_i,y_j) = d\sin\left(\frac{(j-1)c\pi}{N-1}\right), \tag{42}$$

$$M_5(x_i,y_j) = d\sin\left(\frac{(i-1)c\pi}{N-1}\right), \tag{43}$$

where $1 \le i \le M_x+1$, $1 \le j \le M_y+1$, M_x, M_y are the number of nodes of x and y respectively, μ_x is a random number in the range $[M_x/3, 2M_x/3]$, y is a random number in the range $[M_y/3, 2M_y/3]$, σ_x is random number in the range $[1, M_x/4]$, σ_y is random number in the range $[1, M_x/4]$, d is a random number in the range [-2, 2], and c is a random number in the range [1.9, 2.1]. In mutation process, each individual child undergoes mutation with probability *Pmi*. However, for each child that should undergo mutation process, individual curves are mutated with probability *Pmc*. If *Pmi* value is set to 0.5 and *Pmc*

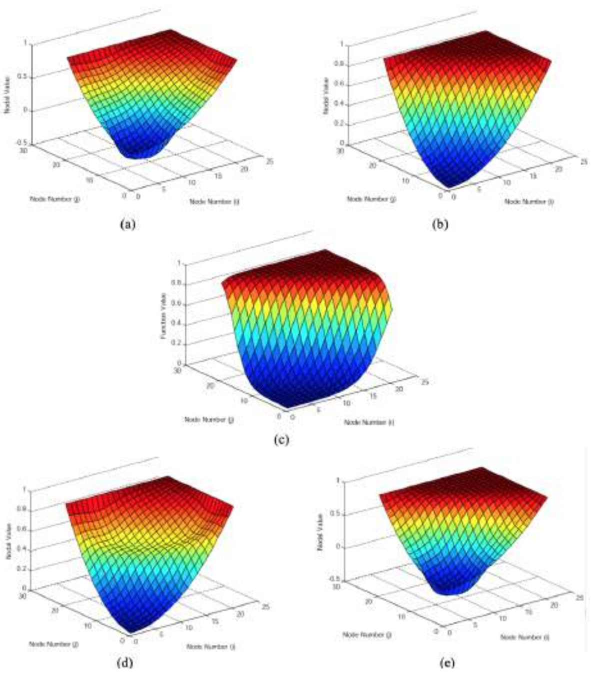


Figure 4: Crossover process: (a) first parent, (b) second parent, (c) crossover function, (d) first child and (e) second child.

value is set to 0.5, then one child out of two children is likely to be mutated, and within that child, $Nc/2$ of the solution curves are likely to be mutated [8]. Figure 5 shows the mutation process between two random parents.

6. Replacement: In the replacement process, to complete the life cycle of the population, the parent population may be replaced by the new one depending on the replacement scheme that we used.

7. Termination: This process is to terminate the CGA when the best fitness is obtained, i.e. the fitness reached to the recurred value, or when the number of generation reached to the maximum value.

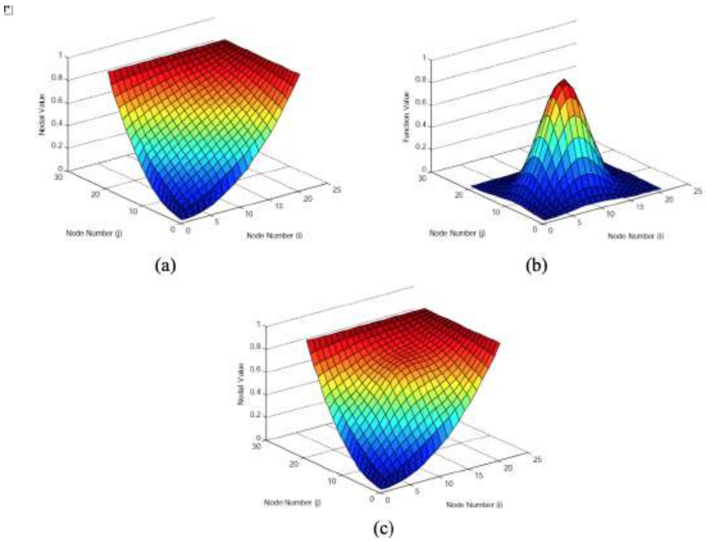
4. Results and illustrations

In this section, we will consider four types of PDEs categorized as Laplace, Poisson, Helmholtz, and nonlinear PDE. The results will be compared for each method's solution in correspondence with the CGA proposed approach.
Problem 1: In this problem, we consider the following Laplace PDE with Neuman conditions [16]

$$\nabla^2 u = 0 \tag{44}$$

with boundary conditions

$$u(0,y)=y, \quad u_x(1,y)=y, \quad u_y(x,0)=x, \quad u_y(x,1)=x.$$



(a) (b)

(c)

Figure 5: Crossover process: (a) first parent, (b) second parent, (c) crossover function, (d) first child and (e) second child.

Problem 2: Consider the Poisson's equation with Dirichlet and Neuman conditions

$$\nabla^2 u = (x^2 + y^2)\exp(xy) \tag{45}$$

with boundary conditions

$$u(0, y) = 0, \quad u_x(2, y) = \exp(2y), \quad u(x, 0) = 1, \quad u(x, 1) = \exp(x).$$

Problem 3: Consider the Helmholtz equation with mixed conditions

$$\nabla^2 u + xu(x, y) = (x + x^2)\exp(y) \tag{46}$$

with boundary conditions

$$u(0, y) = 0, \quad u_x(1, y) = \exp(y), \quad u(x, 0) = x, \quad u_y(x, 1) = x\exp(1).$$

Problem 4: Consider the semilinear elliptic PDE given by

$$\nabla^2 u - (u + 1)^2 = -1 - \exp(2(x + y)) \tag{47}$$

with boundary conditions $u(0, y) = \exp(y)$, $u(1, y) = \exp(y + 1)$, $u(x, 1) = \exp(1 + x)$, $u(x, 2) = \exp(x + 2)$ on $\partial\Omega$, where $\Omega = [0, 1] \times [1, 2]$

To investigate the performance of the proposed CGA method, we analyze each problem by showing the error whether the exact solution exists or not and showing the residual values for each problem, which in return, determines the suitability of the numerical solution for each problem.

Table 1: Error values using CGA for Problem 1

j\i	1	2	3	4	5	6	7	8	9	10	11
1	1.20E-4	1.40E-4	1.30E-4	1.10E-4	8.26E-5	6.77E-5	5.60E-5	5.37E-5	7.44E-5	1.20E-4	1.20E-4
2	1.30E-4	1.49E-4	1.23E-4	1.05E-4	8.49E-5	6.70E-5	5.22E-5	4.59E-5	5.34E-5	1.11E-4	1.18E-4
3	1.00E-4	9.95E-5	9.19E-5	8.53E-5	7.38E-5	6.48E-5	5.20E-5	4.72E-5	2.28E-5	4.43E-5	6.05E-5
4	5.60E-5	5.53E-5	5.87E-5	5.29E-5	5.28E-5	4.51E-5	4.32E-5	4.37E-5	3.19E-5	1.27E-5	1.97E-5
5	5.20E-6	5.92E-6	1.73E-5	2.32E-5	2.91E-5	2.40E-5	2.52E-5	2.57E-5	1.41E-5	3.60E-6	1.82E-6
6	5.43E-5	5.37E-5	3.81E-5	3.07E-5	1.55E-5	1.56E-5	1.07E-5	1.24E-5	1.20E-5	2.94E-5	2.78E-5
7	1.25E-4	1.24E-4	9.33E-5	8.02E-5	5.91E-5	5.29E-5	4.67E-5	4.29E-5	4.24E-5	4.74E-5	4.48E-5
8	1.96E-4	2.05E-4	1.60E-4	1.33E-4	1.10E-4	9.64E-5	8.67E-5	7.75E-5	6.79E-5	6.52E-5	6.14E-5
9	2.37E-4	2.43E-4	2.34E-4	1.79E-4	1.48E-4	1.29E-4	1.09E-4	9.10E-5	8.49E-5	7.49E-5	7.31E-5
10	2.47E-4	2.51E-4	2.75E-4	2.49E-4	1.96E-4	1.61E-4	1.30E-4	1.03E-4	8.42E-5	4.97E-5	6.70E-5
11	2.43E-4	2.48E-4	2.63E-4	2.49E-4	2.07E-4	1.71E-4	1.33E-4	1.04E-4	8.42E-5	6.88E-5	8.54E-5

Table 2: Residual values using CGA for Problem 1

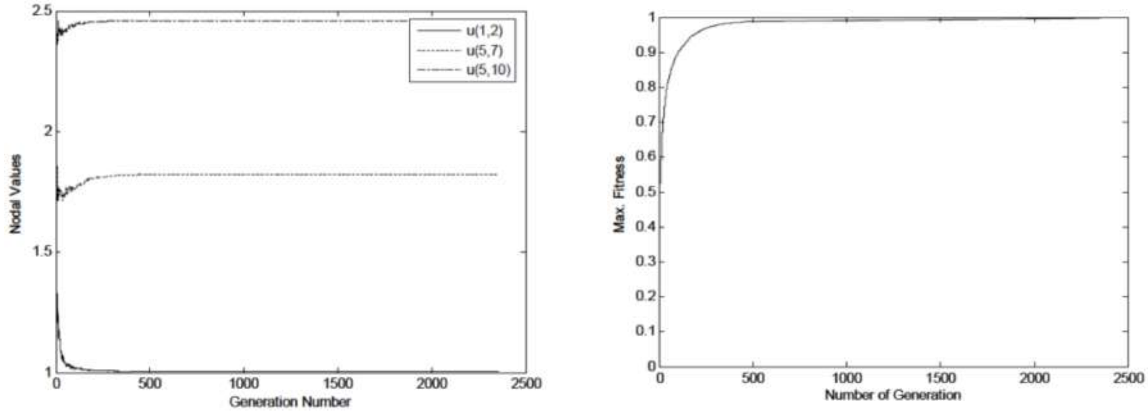
j\i	1	2	3	4	5	6	7	8	9	10	11
1	1.48E-5	3.18E-6	3.52E-6	7.08E-7	3.09E-6	4.86E-7	4.63E-7	1.81E-6	4.44E-6	1.57E-5	1.16E-6
2	3.75E-6	2.50E-5	5.15E-6	4.92E-6	2.90E-6	3.56E-7	3.06E-6	5.73E-6	1.01E-5	2.71E-5	1.84E-5
3	5.07E-6	2.81E-7	1.98E-7	4.52E-6	1.86E-6	5.25E-6	1.99E-7	6.11E-6	2.14E-5	7.40E-6	3.85E-6
4	1.40E-6	2.75E-7	4.36E-6	2.14E-6	2.53E-6	1.05E-6	1.67E-6	6.74E-6	8.53E-6	1.04E-5	1.36E-6
5	1.83E-6	1.09E-7	4.87E-6	6.04E-6	8.04E-6	3.04E-6	4.63E-6	8.00E-6	3.64E-6	2.50E-6	2.00E-6
6	2.51E-6	9.76E-7	1.97E-6	3.03E-6	3.52E-6	1.83E-6	1.65E-6	2.38E-6	5.52E-6	6.74E-6	1.40E-6
7	9.96E-9	5.16E-6	7.27E-6	1.28E-6	5.72E-6	1.53E-6	1.61E-6	1.86E-6	1.61E-7	1.93E-6	1.16E-6
8	3.19E-6	2.43E-5	6.67E-6	3.48E-7	1.56E-6	1.61E-6	4.18E-6	5.46E-6	3.66E-7	2.28E-6	6.32E-7
9	4.73E-6	1.16E-5	1.95E-5	1.20E-5	5.96E-6	6.30E-7	3.03E-8	2.67E-6	5.40E-6	6.68E-6	3.55E-6
10	1.17E-6	2.24E-6	2.58E-5	2.48E-5	4.11E-6	4.87E-6	3.56E-6	5.41E-7	3.76E-6	2.40E-5	2.52E-6
11	4.01E-6	4.09E-6	9.72E-7	6.83E-6	4.73E-6	4.93E-6	8.84E-7	1.46E-6	1.14E-6	1.51E-6	1.75E-5

In the first problem (Laplace equation type), we use 11 × 11 points, where x_0=0 and x_{11}=1, y_0=0 and y_{11}=1. As a result, the errors are shown in Table 1 while the residual values are in Table 2. Figure 6a shows the convergence at selected points and Figure 6b shows the maximum fitness. It is clear that the solution converges in about 1600 generations. In the second problem (Poisson equation type), we use 9 × 11 points, where x_0=0 and x_9=2, y_0=0 and y_{11}=1. As a result, the errors are shown in Table 3 while the residual values are in Table 4. Figure 7a shows the convergence at selected points and Figure 7b shows the maximum fitness. It is clear that the solution converges in about 500 generations. Observing the relations between the errors at each point and the residual values also shows a mutual correspondence.

In the third problem (Helmholtz equation type), we use 11 × 11 points, where x_0=0 and x_{11}=1, $y_0 = 0$ and y_{11}=1. As a result, the errors are shown in Table 5 while the residual values are in Table 6. Figure 8a shows the convergence

Table 3: Error values using CGA for Problem 2

j\i	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	2.80E-4	3.30E-4	2.10E-4	1.80E-4	1.20E-4	2.00E-4	1.81E-4	2.80E-4	0
3	5.80E-4	6.40E-4	3.90E-4	3.30E-4	2.20E-4	4.40E-4	3.16E-4	4.10E-4	0
4	8.70E-4	8.30E-4	5.30E-4	4.30E-4	2.80E-4	6.00E-4	2.06E-5	5.30E-4	0
5	1.11E-3	9.60E-4	6.20E-4	4.60E-4	3.40E-4	6.10E-4	4.50E-4	7.10E-4	0
6	1.30E-3	1.01E-3	6.40E-4	4.60E-4	3.60E-4	5.90E-4	8.50E-4	9.10E-4	0
7	1.44E-3	1.01E-3	6.00E-4	4.40E-4	3.50E-4	5.60E-4	1.16E-3	1.06E-3	0
8	1.50E-3	9.20E-4	5.50E-4	3.80E-4	3.00E-4	5.30E-4	1.32E-3	1.13E-3	0
9	1.38E-3	7.70E-4	4.50E-4	3.20E-4	2.10E-4	4.90E-4	1.18E-3	1.04E-3	0
10	9.10E-4	4.50E-4	2.50E-4	2.10E-4	1.40E-4	3.40E-4	7.50E-4	7.20E-4	0
11	0	0	0	0	0	0	0	0	0

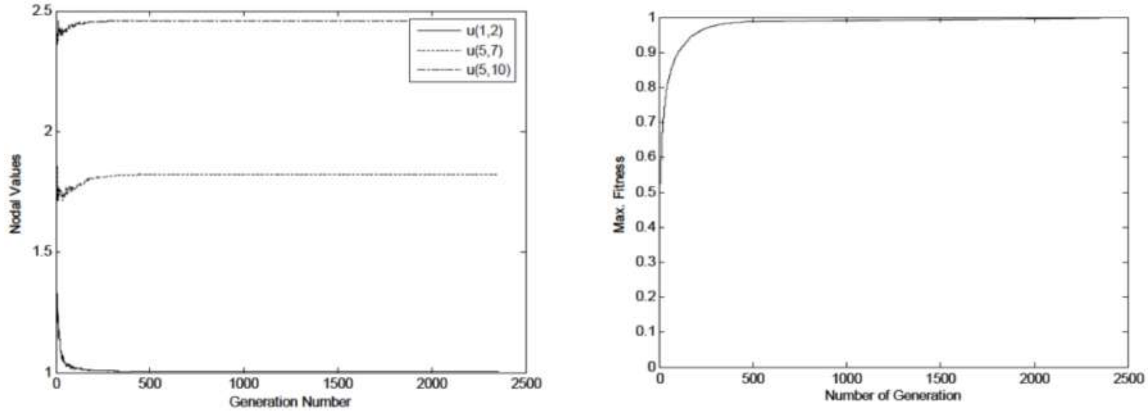


(a) Convergence (b) Maximum Fitness

Figure 6: Convergence and Maximum Fitness for the first problem.

Table 4: Residual values using CGA for Problem 2

j\i	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1.76E-5	2.46E-5	3.11E-6	1.21E-5	5.37E-7	6.61E-6	9.84E-5	7.24E-5	0
3	4.04E-7	6.72E-5	7.80E-6	2.93E-5	1.78E-5	8.75E-5	3.30E-4	3.57E-5	0
4	1.09E-5	4.86E-5	9.09E-6	2.49E-5	3.11E-5	1.14E-4	1.42E-4	1.35E-5	0
5	2.11E-5	4.28E-5	1.24E-5	1.22E-5	1.95E-5	2.88E-5	5.31E-5	4.59E-6	0
6	1.34E-5	3.08E-5	9.89E-6	3.25E-7	1.47E-5	1.35E-5	8.61E-6	8.68E-7	0
7	1.36E-5	3.13E-5	1.86E-5	4.88E-6	1.59E-5	5.88E-5	6.49E-5	1.41E-6	0
8	2.59E-5	7.36E-6	4.01E-6	1.38E-5	2.53E-5	6.36E-5	1.29E-4	3.29E-6	0
9	4.17E-5	3.65E-5	1.38E-5	1.79E-6	6.71E-5	2.58E-5	9.16E-5	1.28E-5	0
10	2.13E-5	2.58E-5	1.47E-5	1.07E-5	2.91E-5	1.12E-5	3.87E-5	1.16E-5	0
11	0	0	0	0	0	0	0	0	0



(a) Convergence (b) Maximum Fitness

Figure 7: Convergence and Maximum Fitness for the second problem.

Table 5: Error values using CGA for Problem 3

j\i	1	2	3	4	5	6	7	8	9	10	11
1	1.75E-1	1.34E-1	3.00E-2	5.97E-2	1.17E-1	1.29E-1	1.04E-1	5.10E-2	2.74E-2	1.12E-1	1.47E-1
2	1.61E-1	1.22E-1	4.46E-2	5.54E-2	1.14E-1	1.28E-1	1.02E-1	4.75E-2	2.98E-2	1.06E-1	1.41E-1
3	1.55E-1	1.25E-1	5.49E-2	5.83E-2	1.20E-1	1.32E-1	1.03E-1	4.82E-2	3.05E-2	1.04E-1	1.34E-1
4	1.48E-1	1.19E-1	4.92E-2	4.88E-2	1.08E-1	1.22E-1	9.53E-2	4.09E-2	3.70E-2	9.30E-2	1.18E-1
5	1.35E-1	1.08E-1	4.03E-2	3.38E-2	8.54E-2	1.01E-1	7.97E-2	3.12E-2	3.73E-2	7.75E-2	9.62E-2
6	1.20E-1	9.49E-2	3.47E-2	1.70E-2	5.81E-2	7.67E-2	6.03E-2	2.23E-2	3.25E-2	6.00E-2	7.39E-2
7	1.02E-1	8.19E-2	3.32E-2	1.13E-3	3.13E-2	5.10E-2	4.13E-2	1.49E-2	2.37E-2	4.26E-2	5.28E-2
8	8.52E-2	6.97E-2	3.19E-2	1.24E-2	8.86E-3	2.89E-2	2.59E-2	1.02E-2	1.35E-2	2.71E-2	3.47E-2
9	7.10E-2	5.94E-2	2.96E-2	2.09E-2	3.14E-3	1.39E-2	1.62E-2	8.87E-3	4.45E-3	1.58E-2	2.16E-2
10	5.65E-2	5.16E-2	3.30E-2	2.12E-2	3.40E-3	1.27E-2	1.66E-2	1.07E-2	1.13E-3	1.12E-2	1.49E-2
11	4.81E-2	4.58E-2	3.90E-2	2.53E-2	6.09E-3	1.00E-2	1.54E-2	1.04E-2	5.96E-4	8.53E-3	1.12E-2

Table 6: Residual values using CGA for Problem 3

j\i	1	2	3	4	5	6	7	8	9	10	11
1	3.04E-2	1.91E-2	1.67E-2	7.73E-3	6.93E-3	4.70E-3	4.37E-3	5.68E-3	6.60E-3	8.87E-3	1.27E-2
2	1.37E-2	2.29E-3	3.67E-3	8.35E-4	3.96E-4	4.12E-4	1.70E-4	3.74E-4	1.03E-3	7.46E-4	9.81E-3
3	1.23E-2	7.05E-3	1.04E-2	9.72E-3	1.05E-2	8.21E-3	4.95E-3	5.39E-3	4.45E-3	5.30E-3	1.04E-2
4	1.23E-2	5.60E-3	5.63E-3	5.81E-3	7.86E-3	6.43E-3	5.55E-3	4.31E-3	3.81E-3	3.11E-3	6.00E-3
5	1.12E-2	4.31E-3	6.18E-3	2.77E-3	4.54E-3	4.03E-3	3.28E-3	2.27E-3	4.17E-3	1.86E-3	3.09E-3
6	9.15E-3	3.36E-3	3.30E-3	9.27E-4	2.13E-3	2.87E-3	2.51E-3	2.35E-3	3.55E-3	1.59E-3	1.59E-3
7	6.13E-3	2.60E-3	2.36E-3	1.26E-3	2.44E-3	2.35E-3	2.32E-3	2.20E-3	1.97E-3	1.77E-3	1.75E-3
8	3.87E-3	1.98E-3	3.12E-3	2.06E-3	4.29E-3	2.89E-3	2.71E-3	2.64E-3	2.23E-3	2.03E-3	2.14E-3
9	5.32E-3	2.44E-3	4.59E-3	4.03E-3	4.81E-3	5.43E-3	4.11E-3	3.39E-3	3.49E-3	2.70E-3	2.60E-3
10	4.48E-3	4.45E-4	2.03E-3	1.15E-3	1.21E-3	1.71E-3	1.29E-3	1.25E-3	9.98E-4	4.40E-4	1.97E-3
11	9.31E-3	5.27E-3	7.95E-3	5.17E-3	3.59E-3	2.49E-3	1.39E-3	1.29E-3	1.18E-3	2.21E-3	4.12E-3

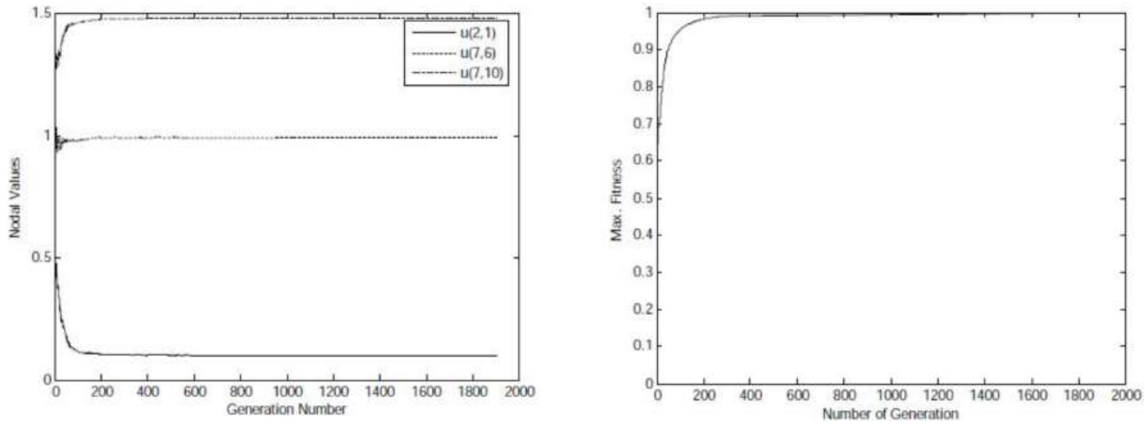
at selected points and Figure 8b shows the maximum fitness. It is clear that the solution converges in about 4000 generations.

In the fourth problem (nonlinear equation type), we use 11 × 11 points, where x_0=0 and x_{11}=1, y_0=1 and y_{11}=2. As a result, the errors are shown in Table 7 while the residual values are in Table 8. Figure 9a shows the convergence at selected points and Figure 9b shows the maximum fitness. It is clear that the solution converges in about 400 generations. Observing the relations between the errors at each point and the residual values, again shows a mutual correspondence.

Now, in order to investigate the potential of the proposed CGA approach, the previous four linear partial differential equation boundary value problems have also been solved using the FDM and FEM for a comparison purpose. Solutions of the FEM and FDM are summarized in Table 9 and Table 10 respectively, while the solutions of the CGA are summarized in Table 11.

Table 7: Error values using CGA for Problem 4

j\i	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	7.28E-5	1.20E-4	1.50E-4	1.70E-4	1.30E-4	7.84E-5	6.00E-5	2.20E-4	7.55E-5	0
3	0	1.20E-4	2.20E-4	2.70E-4	2.90E-4	2.80E-4	2.00E-4	1.41E-4	2.07E-4	3.10E-4	0
4	0	1.50E-4	2.70E-4	3.50E-4	3.90E-4	4.20E-4	4.00E-4	3.40E-4	1.91E-5	2.09E-5	0
5	0	1.60E-4	3.00E-4	3.90E-4	4.40E-4	4.60E-4	4.30E-4	3.50E-4	2.60E-4	1.50E-4	0
6	0	1.40E-4	3.00E-4	4.10E-4	4.60E-4	4.90E-4	4.80E-4	4.40E-4	3.40E-4	2.00E-4	0
7	0	1.20E-4	3.10E-4	4.10E-4	4.50E-4	4.90E-4	4.90E-4	4.60E-4	3.90E-4	2.50E-4	0
8	0	8.93E-5	3.47E-5	3.40E-4	3.90E-4	4.40E-4	4.40E-4	4.40E-4	3.70E-4	2.50E-4	0
9	0	4.20E-4	4.35E06	5.16E-5	3.20E-4	3.60E-4	3.80E-4	3.80E-4	3.40E-4	2.40E-4	0
10	0	8.74E-5	2.80E-4	7.00E-5	1.90E-4	2.30E-4	2.50E-4	2.50E-4	2.30E-4	1.70E-4	0
11	0	0	0	0	0	0	0	0	0	0	0

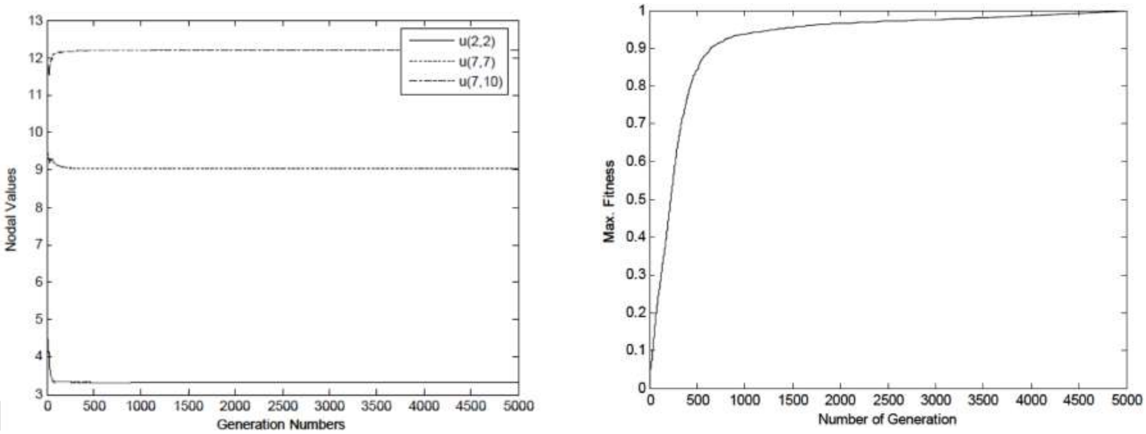


(a) Convergence (b) Maximum Fitness

Figure 8: Convergence and Maximum Fitness for the third problem.

Table 8: Residual values using CGA for Problem 4

j\i	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	4.15E-5	4.02E-5	2.22E-5	7.90E-5	2.63E-4	5.15E-4	1.62E-3	2.78E-3	1.42E-3	0
3	0	1.67E-5	8.76E-5	7.68E-5	4.30E-5	1.60E-4	5.31E-4	3.08E-3	5.65E-3	3.37E-3	0
4	0	4.62E-5	7.30E-5	9.17E-5	8.98E-5	4.40E-4	7.47E-4	2.47E-3	3.65E-3	1.72E-3	0
5	0	2.27E-5	1.14E-4	8.86E-5	1.40E-4	1.98E-4	2.35E-4	7.06E-4	6.47E-4	3.94E-4	0
6	0	1.80E-4	5.45E-5	7.30E-5	7.42E-5	1.07E-4	1.57E-4	1.28E-4	9.41E-5	2.79E-4	0
7	0	3.97E-4	9.49E-4	2.18E-4	1.30E-4	9.52E-5	1.15E-4	1.22E-4	1.95E-4	5.47E-5	0
8	0	1.99E-3	1.67E-3	1.20E-3	3.04E-4	1.05E-4	1.20E-4	1.50E-4	1.38E-4	4.32E-5	0
9	0	3.27E-3	1.98E-3	2.14E-3	4.00E-4	7.42E-5	1.06E-4	8.29E-5	2.76E-4	6.09E-5	0
10	0	9.08E-4	1.74E-3	7.81E-4	2.79E-4	6.72E-5	3.43E-5	2.53E-5	4.99E-5	5.19E-5	0
11	0	0	0	0	0	0	0	0	0	0	0



(a) Convergence (b) Maximum Fitness

Figure 9: Convergence and Maximum Fitness for the fourth problem.

Table 9: Numerical results using the FEM approach

#	**Type**	**Average error**	$\|\|Error\|\|_\infty$	$\|\|u\|\|_\infty$	$\|\|Exact\|\|_\infty$	**Max. error**	**Min. error**
problem 1	Laplace	2.19E-1	5.11	5.38	7.90	2.21E-1	2.16E-1
problem 2	Poisson	4.92E-4	3.22E-2	8.43E+1	8.43E+1	6.38E-3	1.04E-8
problem 3	Helmholtz	1.89E-4	5.26E-3	2.85E+1	28.51E+1	9.65E-4	4.49E-7
problem 4	Nonlinear	1.79E-2	1.50E-1	8.11E+1	8.12	4.99E-2	3.70E-3

Table 10: Numerical results using the FDM approach

#	**Type**	**Average error**	$\|\|Error\|\|_\infty$	$\|\|u\|\|_\infty$	$\|\|Exact\|\|_\infty$	**Max. error**	**Min. error**
problem 1	Laplace	2.30E-1	2.53	5.90	3.85	2.30E-1	2.30E-1
problem 2	Poisson	1.45E-2	2.17E-1	2.45E+1	2.44E+1	6.29E-2	7.58E-4
problem 3	Helmholtz	7.85E-3	1.42E-1	1.18E+1	1.18E+1	2.64E-2	6.81E-4
problem 4	Nonlinear	1.39E-1	1.18	7.33	7.26	2.88E-1	7.15E-2

As clearly seen from these tables, the proposed CGA approach outperforms the well known FEM and FDM in different places focusing on the average error.

5. Conclusion

In this paper, continuous genetic algorithms have been implemented for the solution of PDE's/PVB's given in the form of Laplace, Poisson, Helmholtz, and nonlinear. To investigate the performance of the proposed CGA technique, the four different problems were solved by the CGA and then compared their results with the results obtained by the FDM and FEM. Thus, it was concluded that the CGA performs its operation with different advantages; being a parallel type method (reduction of computational cost), requires a minimal problem knowledge, and being, in a sense, non-iterative technique. In addition to that, one can guess the suitability of the numerical solution to a given problem by the residual value; that is, whenever the residual value converges to zero the error converges to zero. Based on the analysis of the illustrative problems, it is shown clearly that the proposed CGA technique operates with a high potential and is suitable for solving different types of PDE's/PVB's with different advantages as presented.

Table 11: Numerical results using the CGA approach

#	**Type**	**Average error**	$\|\|Error\|\|_\infty$	$\|\|u\|\|_\infty$	$\|\|Exact\|\|_\infty$	**Max. error**	**Min. error**
problem 1	Laplace	9.16E-5	5.165E-6	1.23E-3	5.47E-5	2.75E-4	1.81E-6
problem 2	Poisson	6.03E-4	3.37E-5	5.77E-3	4.51E-4	1.49E-3	2.05E-5
problem 3	Helmholtz	3.50E-4	1.74E-5	4.01E-3	2.88E-4	9.25E-4	2.65E-6
problem 4	Nonlinear	2.72E-4	6.55E-4	2.58E-3	9.48E-3	4.88E-4	4.34E-6

References

[1] O. Abu Arqub, Z. Abo-Hammour, S. Momani, *Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems*, Applied Mathematics and Information Sciences, 8 (2014), 414-432.

[2] O. Abu Arqub, Z. Abo-Hammour, R. Al-Badarneh, S. Momani, *A reliable analytical method for solving higher-order initial value problems*, Discrete Dynamics in Nature And Society, 2013, 1-12.

[3] O. Abu Arqub, Z. Abo-Hammour, S. Momani, N. Shawagfeh, *Solving singular two-point boundary value problems using continuous genetic algorithm*, Abstract And Applied Analysis, 2012, 1-25.

[4] Z. Abo-Hammour, O. Alsmadi, S. Momani, O. Abu Arqub, *A genetic algorithm approach for prediction of linear dynamical systems*, Mathematical Problems in Engineering, vol. 2013 Article ID831657, 2013, 1-12.

[5] Z. Abo-Hammour, O. Alsmadi, A. Al-Smadi, M. Zaqout, M. Saraireh, *ARMA model order and parameter estimation using genetic algorithms*, Mathematical and Computer Modelling of Dynamical Systems, 18 (2012), 201-221.

[6] Z. Abo-Hammour, A. Asasfeh, A. Al-Smadi, O. Alsmadi, *A novel continuous genetic algorithm for the solution of optimal control problems*, Optimal Control Applications And Methods, 32 (2011a), 414-432.

[7] Z. Abo-Hammour, O. Alsmadi, S. Bataineh, M. Al-Omari, A. Affect, *Collision-free cartesian path planning of robot manipulators*, International Journal of Advanced Robotic Systems, 8 (2011b), 14-36.

[8] Z. Abo-Hammour, *Advanced continuous genetic algorithms and their applications in the motion planning of robotic manipulators and the numerical solution of boundary value problems*, Ph.D. Thesis, Quiad-Azam University, Pakistan, 2002.

[9] R.B. Albadarneh, N.T. Shawagfeh, Z.S. Abo-Hammor, *Numerical solution of semilinear elliptic equations using 13-points FDM*, International Mathematical Forum, 5 (2010), 527-536.

[10] R.B. Albadarneh, N.T. Shawagfeh, Z.S. Abo-Hammor, *General (n+1)-explicit finite difference formulas with proof*, Appl. Math. Sci., 6 (2012), 995-1009.

[11] A. K. Alomari, M. M. S. Noorani, R. Nazar, *Comparison between the homotopy analysis method and homotopy perturbation method to solve coupled Schrodinger-KdV equation*, Journal of Applied Mathematics and Computing, 31 (2009), 1-12.

[12] A. K. Alomari, M. M. S. Noorani, R. Nazar, *On the homotopy analysis method for the exact solutions of Helmholtz equation*, Chaos, Solitons & Fractals, 41 (2009), 1873-1879.

[13] A.K. Alomari, M.S.M. Noorani, R. Nazar, *Explicit series solutions of some linear and nonlinear Schrodinger equations via the homotopy analysis method*, Communications in Nonlinear Science and Numerical Simulation, 14 (2009), 1196-1207.

[14] O. Alsmadi, Z. S. Abo-Hammour, A. M. Al-Smadi, D. I. Abu-Al-Nadi. *Genetic algorithm approach with frequency selectivity for model order reduction of MIMO systems*, Mathematical and Computer Modelling of Dynamical Systems, 17 (2011), 163-181.

[15] I. Aslan, *Application of the exp-function method to nonlinear lattice differential equations for multiwave and rational solutions*, Mathematical Methods in the Applied Sciences, 34 (2011), 1707-1710.

[16] R. L. Burden, J.D. Faires, *Numerical analysis*, seventh ed., Brooks/Cole, Pacific Grove, CA, 2001.

[17] S.C. Chapra, R.P. Canale, *Numerical methods for engineers*, third ed., McGraw-Hill, New York, 1998.

[18] D. Coloton, R. Kress, *Inverse acoustic and electromagnetic scattering theory*, 2nd ed., Springer-Verlag, Germany, 1998.

[19] T. Dokken, T. Lyche, *A divided difference formula for the error in hermite interpolation*, BIT, 19 (1979), 539-540.

[20] W. Forst, *Interpolation undnumerische differentiation*, J. Approx. Theory, 39 (1983), 118-131.

[21] C.F. Gerald, P.O. Wheatley, *Applied numerical analysis*, fourth ed., Addison Wesley Pub. Co., Reading, MA, (1989).

[22] L.P. Grabar, *Numerical differentiation by means of Chebyshev polynomials orthonormalized on a system of equidistant points*, Zh. Vychisl, Mat. i Mat. Fiz., 7 (1967), 1375-1379.

[23] M.W. Gutowski, *Smooth genetic algor*, J. Phys. A: Math. Gen., 27 (1994), 7893-7904.

[24] R.W. Hamming, *Numerical methods for scientists and engineers*, McGraw Hill, NewYork, 1962.

[25] W. H. Hayt, *Engineering electromagnetics*, McGraw-Hill, 1981.

[26] M. Hazewinkel, *Encyclopedia of mathematics*, Springer, Germany, 2001.

[27] D. He, K. Pan, *An order optimal regularization method for the Cauchy problem of a Laplace equation in an annulus domain*, Applied Mathematical Modelling, 39 (2015), 3063-3074.

[28] J. Fish, T. Belytschko, *A first course in finite elements*, John Wiley & Sons Ltd, 2007.

[29] A. Jaskiewicz, *On a continuous solution to the Bellman-Poisson equation in stochastic games*, Journal of Optimization Theory And Applications, 145 (2010), 451-458.

[30] H. Jeffreys, B. S. Jeffreys, *Methods of mathematical physics*, 3rd ed. Cambridge, England, Cambridge University Press, 2000.

[31] L. Jianping, *General explicit difference formulas for numerical differentiation*, Journal of Computational and Applied Mathematics, 183 (2005), 29-52.

[32] K. De. Jong, *A 10 year perspective*, in Proc. Int. Conf. Genetic Algorithms Their Applications, (1985), 169-177.

[33] G. Katti, N. DasGupta, A. DasGupta, *Threshold voltage model for mesa-isolated small geometry fully depleted SOI MOSFETs based on analytical solution of 3-D Poisson's equation*, IEEE Transactions On Electron Devices, 51 (2004), 1169-1177.

[34] E. Kreyzig, *Advanced engineering mathematics*, seventh ed., Wiley, NewYork, 1994.

[35] T.N. Krishnamurti, L. Bounoua, *An introduction to numerical weather prediction technique techniques*, CRC Press, Boca Raton, FL, 1995.

[36] B.I. Kvasov, *Numerical differentiation and integration on the basis of interpolation parabolic splines*, Chisl.Metody Mekh. Sploshn.Sredy, 14 (1983), 68-80.

[37] L. L. Lapin, *Probability and statistics for modern engineering*, Second Edition, PWS-KENT, 1990.

[38] J.H. Mathews, K.D. Fink, *Numerical methods using MATLAB*, third ed., Prentice Hall, Inc., Englewood Cliffs, NJ, 1999.

[39] L. Lussardi, *On a Poisson's equation arising from magnetism*, Discrete And Continuous Dynamical Systems-Series S, 8 (2015), 769-772.

[40] R. McOwen, *Partial differential equations*, Prentice Hall, Inc, 1996.

[41] K. W. Morton, D. F. Mayers, *Numerical solution of partial differential equations an introduction*, Second Edition,Cambridge University Press, 2005.

[42] Polyanin and A. V. Manzhirov, *Handbook of mathematics for engineers and scientists*, Chapman & Hall/CRC Press, 2006.

[43] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical recipes in FORTRAN: the art of scientific computing*, 2nd ed. Cambridge, England Cambridge University Press, 1992.

[44] V. D. Radulescu, *Qualitative analysis of nonlinear elliptic partial differential equations: monotonicity*, Analytic, Variational Methods, Hindawi Publishing Corporation, New York, 2008.

[45] R. Seroul, *Programming for mathematicians*, Berlin, Springer Verlag, 2000.

[46] P. P. Silvester, R. L. Ferrari, *Finite elements for electrical engineers*, 3rd ed. Cambridge University Press, Cambridge, U.K, (1996).

[47] P. Silvester, *Numerical formation of finite difference operators*, IEEE Trans. Micro wave Theory Technol. MTT-18, (1970), 740-743.

[48] E. Tadmor, *A review of numerical methods for nonlinear partial differential equations*, Bulletin (New Series) Of The American Mathematical Society, 49 (2012), 507-554.

[49] E. A. Volkov, *Modified combined grid method for solving the Dirichlet problem for the Laplace equation on a rectangular parallelepiped*, Computational Mathematics And Mathematical Physic, 50 (2010), 274-284.

[50] K.B. Wolf , E.V. Kurmyshev, *Squeezed states in Helmholtz optics*, Physical Review, 47 (1993), 3365-3370.

[51] A. Yildirim, Z. Pnar, *Application of the exp-function method for solving nonlinear reaction diffusion equations arising in mathematical biology*, Computers and Mathematics with Applications, 60 (2010), 1873-1880.

[52] W. Zhang, *The extended tanh method and the exp-function method to solve a kind of nonlinear heat equation*, Mathematical Problems in Engineering, Article ID 935873, (2010), 1-12.

Accepted: 28.02.2019