Scientia Iranica D (2017) 24(6), 3181–3192

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.scientiairanica.com

New high-accuracy non-polynomial spline group explicit iterative method for two-dimensional elliptic boundary value problems

J. Goh* and **N.H.M. Ali**

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia.

Received 9 September 2015; received in revised form 24 May 2016; accepted 20 August 2016

KEYWORDS
Non-polynomial spline;
Finite difference method;
Group explicit iterative method;
Elliptic partial differential equation.

Abstract. In this paper, we propose a new high-accuracy method based on non-polynomial spline for the numerical solution to two-dimensional elliptic partial differential equations. Using a non-polynomial spline approximation in x-direction and central difference in y-direction, we obtain a new nine-point compact finite-difference formulation. A four-point Group Explicit (GE) iterative scheme with an acceleration tool is then applied to the obtained system. The formulation procedure is presented in detail. The efficiency of the proposed method is then illustrated by some test problems. The numerical results are found to be in good agreement with the exact solutions.

1. Introduction

Equilibrium problems in two or higher dimensions often lead to the elliptic partial differential equations. These equations arise very frequently in describing velocity potentials, stationary distribution of temperatures, potential flows, and structural mechanics. Thus, solving this type of equation has been of interest to many researchers [1-5]. We consider the two-dimensional elliptic partial differential equation of the form:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=A(x,y)\frac{\partial u}{\partial x}+B(x,y)\frac{\partial u}{\partial y}+g(x,y),$$

$$(x,y)\in\Omega, \qquad (1)$$

defined in solution domain $\Omega=\{(x,y):0<x,y<1\}$ with boundary $\partial\Omega$, where $A(x,y)>0$ and $B(x,y)>0$ in Ω. The corresponding Dirichlet boundary conditions are prescribed by:

$$u(x,y)=\psi(x,y), \qquad (x,y)\in\partial\Omega. \qquad (2)$$

Assume that the boundary conditions are given with sufficient smoothness to maintain the order of accuracy in the numerical method under consideration.

The construction of group iterative methods in solving the elliptic partial differential equation with promising results and improved execution timings has been greatly observed since the 1980s [6-12]. The methods were formulated using a combination of skewed finite-difference approximations together with the centred-difference approximation that resulted in schemes with better rates of convergence than the existing iterative methods available in literature. However, one of the weaknesses of these formulations is that the formulas are based purely on finite-difference discretization which enables the solutions to be obtained only at certain intersection points of the grid lines in the solution domain.

The application of splines to solving differential equations has been an active area of research over the

*. *Corresponding author. Tel.: +604 653 3924;*
E-mail address: joangoh.usm@gmail.com (J. Goh)

doi: 10.24200/sci.2017.4487

last few decades. In 1968, Bickley [13] originated an idea to obtain better accuracy for a linear ordinary differential equation by using cubic splines method. Following this, Albasiny and Hoskins [14] applied the cubic spline interpolation to solve a two-point boundary value problem. At about the same time, Fyfe [15] examined the method suggested by Bickley [13] and carried out the error analysis. Fyfe concluded that spline method is better than the usual finite-difference method as the spline method has the flexibility to get the solution at any point in the domain with more accurate results. Due to its simplicity, many researchers started considering spline as one of the approximation tool to obtain accurate numerical solutions [16-19]. Recently, Ding et al. [20] and Gopal et al. [21] have studied non-polynomial spline methods for the numerical solution to one-dimensional hyperbolic problem. More recently, Jha and Mohanty [22] have formulated the solution to nonlinear second-order boundary value problems by using quintic non-polynomial spline. To the best of the authors' knowledge, no high-accuracy non-polynomial spline method has been investigated on two-dimensional elliptic partial differential equations. Since the spline method has the flexibility to produce approximations at any point in the domain with high-accuracy solutions, there has been an interest to formulate group iterative schemes in hybrid with splines in solving the elliptic partial differential equations. Goh and Ali [23] managed to derive a new method, namely the Spline Explicit Group (SEG) iterative method, which incorporates cubic spline with group iterative scheme for solving the elliptic equation with the promising results. However, with the emergence of newer types of splines with more favorable properties, it would be a worthwhile effort to investigate the application of highly advanced types of splines to the group schemes as a means to further improve the performance of the methods.

In this paper, we aim to discuss the formulation of a new numerical method which incorporates a non-polynomial spline into combination with a group explicit iterative scheme for solving a two-dimensional elliptic partial differential equation. We will present the proposed method as follows. In the next section, we discuss the non-polynomial spline approximations. In Section 3, the numerical scheme of the solution to the singular problem based on the non-polynomial spline approximation in x-direction and central difference approximation in y-direction will be elaborated in detail. The formulation of the non-polynomial spline group explicit iterative method will be discussed and the complexity of computation will be analyzed in Section 4. The performance of method will be investigated via few test problems in Section 5. Finally, the discussion and concluding remarks will be given in Sections 6 and 7, respectively.

2. The non-polynomial spline approximation

Let solution domain $\Omega = [0,1] \times [0,1]$ be divided into $N_x \times N_y$ mesh with spatial step size $h = 1/N_x > 0$ in x-direction and $k = 1/N_y > 0$ in y-direction, respectively, where N_x and N_y are positive integers. The mesh ratio parameter is denoted by $\lambda = (k/h) > 0$. Grid points (x_l, y_m) are defined by $x_l = lh$ and $y_m = mk$, $l = 0, 1, \cdots, N_x$, $m = 0, 1, \cdots, N_y$. Notations $u_{l,m}$ and $U_{l,m}$ are represented as the exact and approximation solutions of $u(x,y)$ at grid point (x_l, y_m), respectively. At grid point (x_l, y_m), differential equation (1) can be written as follows:

$$U_{xxl,m} + U_{yyl,m} = f(x_l, y_m, U_{l,m}, U_{xl,m}, U_{yl,m}) \equiv F_{l,m}, \quad (3)$$

where:

$$f(x,y,U,U_x,U_y) = A(x,y)U_x + B(x,y)U_y + g(x,y).$$

In this paper, a non-polynomial spline approximation is used to approximate the solution. Non-polynomial spline, $S_m(x)$, is a function of class $C^2[0,1]$, which interpolates value $U_{l,m}$ at grid point (x_l, y_m) at each mth mesh row and is given by:

$$S_m(x) = a_{l,m} + b_{l,m}(x - x_l) + c_{l,m}\sin\omega(x - x_l) + d_{l,m}\cos\omega(x - x_l), \qquad x_l \le x \le x_{l+1}, \quad (4)$$

where $a_{l,m}$, $b_{l,m}$, $c_{l,m}$, and $d_{l,m}$ are constants and ω is a free parameter.

The derivatives of non-polynomial spline $S_m(x)$ can be obtained as follows:

$$S'_m(x) = b_{l,m} + \omega c_{l,m}\cos\omega(x - x_l) - \omega d_{l,m}\sin\omega(x - x_l), \quad (5)$$

$$S''_m(x) = -\omega^2[c_{l,m}\sin\omega(x - x_l) + d_{l,m}\cos\omega(x - x_l)]. \quad (6)$$

In order to derive the expression for the coefficients of Eq. (4) in terms of $U_{l,m}$, $U_{l+1,m}$, $M_{l,m}$, and $M_{l+1,m}$, we denote:

$$S_m(x_l) = U_{l,m}, \qquad S_m(x_{l+1}) = U_{l+1,m},$$

$$S''_m(x_l) = M_{l,m}, \qquad S''_m(x_{l+1}) = M_{l+1,m}.$$

From algebraic manipulation, we can obtain:

$$a_{l,m} = U_{l,m} + \frac{M_{l,m}}{\omega^2}, \qquad c_{l,m} = \frac{M_{l,m}\cos\theta - M_{l+1,m}}{\omega^2 \sin\theta},$$

$$b_{l,m} = \frac{U_{l+1,m} - U_{l,m}}{h} + \frac{M_{l+1,m} - M_{l,m}}{\omega\theta},$$

$$d_{l,m} = -\frac{M_{l,m}}{\omega^2},$$

where $\theta = \omega h$. By substituting $x = x_l$ and the constants into Eq. (5), we obtain:

$$m_{l,m} = S'_m(x_l) = U_{xl,m} = \frac{U_{l+1,m} - U_{l,m}}{h} - h[\alpha M_{l+1,m} + \gamma M_{l,m}], \tag{7}$$

where:

$$\alpha = \frac{1}{\theta^2}(\theta \csc\theta - 1), \qquad \gamma = \frac{1}{\theta^2}(1 - \theta\cot\theta).$$

Replacing h by $-h$, it gives:

$$m_{l,m} = S'_m(x_l) = \frac{U_{l,m} - U_{l-1,m}}{h} + h[\alpha M_{l-1,m} + \gamma M_{l,m}]. \tag{8}$$

Combining both Eqs. (7) and (8), the following approximation can be obtained:

$$m_{l,m} = S'_m(x_l) = \frac{U_{l+1,m} - U_{l-1,m}}{2h} - \frac{\alpha h}{2}[M_{l+1,m} - M_{l-1,m}]. \tag{9}$$

Further, we have:

$$m_{l+1,m} = S'_m(x_{l+1}) = \frac{U_{l+1,m} - U_{l,m}}{h} + h[\alpha M_{l,m} + \gamma M_{l+1,m}], \tag{10}$$

$$m_{l-1,m} = S'_m(x_{l-1}) = \frac{U_{l,m} - U_{l-1,m}}{h} - h[\alpha M_{l,m} + \gamma M_{l-1,m}]. \tag{11}$$

By using the continuity of the first derivative at (x_l, y_m), which is $S'_m(x_l^+) = S'_m(x_l^-)$, the following relation can be obtained:

$$\frac{U_{l+1,m} - 2U_{l,m} + U_{l-1,m}}{h^2} = \alpha M_{l+1,m} + 2\gamma M_{l,m} + \alpha M_{l-1,m}. \tag{12}$$

It is worthwhile to notice that when $\omega \to 0$, that $\theta \to 0$ and $(\alpha, \gamma) \to (1/6, 1/3)$, then the relation reduces to ordinary cubic spline relation:

$$U_{l+1,m} - 2U_{l,m} + U_{l-1,m} = \frac{h^2}{6}(M_{l+1,m} + 4M_{l,m} + M_{l-1,m}).$$

The following approximations are considered:

$$\overline{U}_{yl,m} = \frac{U_{l,m+1} - U_{l,m-1}}{2k} = U_{yl,m} + \frac{k^2}{6}U_{03} + O(k^4), \tag{13a}$$

$$\overline{U}_{yl+1,m} = \frac{U_{l+1,m+1} - U_{l+1,m-1}}{2k} = U_{yl+1,m} + \frac{k^2}{6}U_{03} + \frac{k^2 h}{6}U_{13} + O(k^2h^2), \tag{13b}$$

$$\overline{U}_{yl-1,m} = \frac{U_{l-1,m+1} - U_{l-1,m-1}}{2k} = U_{yl-1,m} + \frac{k^2}{6}U_{03} - \frac{k^2 h}{6}U_{13} + O(k^2h^2), \tag{13c}$$

$$\overline{U}_{yyl,m} = \frac{U_{l,m+1} - 2U_{l,m} + U_{l,m-1}}{k^2} = U_{yyl,m} + \frac{k^2}{12}U_{04} + O(k^4), \tag{14a}$$

$$\overline{U}_{yyl+1,m} = \frac{U_{l+1,m+1} - 2U_{l+1,m} + U_{l+1,m-1}}{k^2} = U_{yyl+1,m} + \frac{k^2}{12}U_{04} + \frac{k^2 h}{12}U_{14} + O(k^2h^2), \tag{14b}$$

$$\overline{U}_{yyl-1,m} = \frac{U_{l-1,m+1} - 2U_{l-1,m} + U_{l-1,m-1}}{k^2} = U_{yyl-1,m} + \frac{k^2}{12}U_{04} - \frac{k^2 h}{12}U_{14} + O(k^2h^2), \tag{14c}$$

$$\overline{m}_{l,m} = \frac{U_{l+1,m} - U_{l-1,m}}{2h} = m_{l,m} + \frac{h^2}{6}U_{30} + O(h^4), \tag{15a}$$

$$\overline{m}_{l+1,m} = \frac{3U_{l+1,m} - 4U_{l,m} + U_{l-1,m}}{2h} = m_{l+1,m} - \frac{h^2}{3}U_{30} - O(h^3), \tag{15b}$$

$$\overline{m}_{l-1,m} = \frac{-3U_{l-1,m} + 4U_{l,m} - U_{l+1,m}}{2h} = m_{l-1,m} - \frac{h^2}{3}U_{30} + O(h^3), \tag{15c}$$

where:

$$W_{ab} = \frac{\partial^{a+b} W(x_l, y_m)}{\partial x^a \partial y^b}, \qquad W = U, D \text{ and } g.$$

For the derivatives of $S_m(x)$, we consider:

$$\overline{M}_{l,m} = -\overline{U}_{yyl,m} + \overline{F}_{l,m}, \quad (16a)$$

$$\overline{M}_{l+1,m} = -\overline{U}_{yyl+1,m} + \overline{F}_{l+1,m}, \quad (16b)$$

$$\overline{M}_{l-1,m} = -\overline{U}_{yyl-1,m} + \overline{F}_{l-1,m}, \quad (16c)$$

$$\hat{m}_{l,m} = \frac{U_{l+1,m} - U_{l-1,m}}{2h} - \frac{\alpha h}{2}\left[\overline{M}_{l+1,m} - \overline{M}_{l-1,m}\right], \quad (17a)$$

$$\hat{m}_{l+1,m} = \frac{U_{l+1,m} - U_{l,m}}{h} + h\left[\alpha\overline{M}_{l,m} + \gamma\overline{M}_{l+1,m}\right], \quad (17b)$$

$$\hat{m}_{l-1,m} = \frac{U_{l,m} - U_{l-1,m}}{h} - h\left[\alpha\overline{M}_{l,m} + \gamma\overline{M}_{l-1,m}\right], \quad (17c)$$

and:

$$\overline{F}_{l,m} = f(x_l, y_m, U_{l,m}, \overline{U}_{xl,m}, \overline{U}_{yl,m}), \quad (18a)$$

$$\overline{F}_{l+1,m} = f(x_{l+1}, y_m, U_{l+1,m}, \overline{U}_{xl+1,m}, \overline{U}_{yl+1,m}), \quad (18b)$$

$$\overline{F}_{l-1,m} = f(x_{l-1}, y_m, U_{l-1,m}, \overline{U}_{xl-1,m}, \overline{U}_{yl-1,m}), \quad (18c)$$

$$\hat{F}_{l,m} = f(x_l, y_m, U_{l,m}, \hat{U}_{xl,m}, \overline{U}_{yl,m}), \quad (19a)$$

$$\hat{F}_{l+1,m} = f(x_{l+1}, y_m, U_{l+1,m}, \hat{U}_{xl+1,m}, \overline{U}_{yl+1,m}), \quad (19b)$$

$$\hat{F}_{l-1,m} = f(x_{l-1}, y_m, U_{l-1,m}, \hat{U}_{xl-1,m}, \overline{U}_{yl-1,m}). \quad (19c)$$

Let:

$$\eta_{l,m} = \left(\frac{\partial f}{\partial U_x}\right)_{l,m}. \quad (20)$$

With the help of approximations (13) and (15), from Eqs. (18a)-(18c), we obtain:

$$\overline{F}_{l,m} = F_{l,m} + \frac{h^2}{6}U_{30}\eta_{l,m} + O(h^4 + k^2), \quad (21a)$$

$$\overline{F}_{l+1,m} = F_{l+1,m} - \frac{h^2}{3}U_{30}\eta_{l,m} + O(-h^3 + k^2 + k^2h + k^2h^2), \quad (21b)$$

$$\overline{F}_{l-1,m} = F_{l-1,m} - \frac{h^2}{3}U_{30}\eta_{l,m} + O(h^3 + k^2 - k^2h + k^2h^2). \quad (21c)$$

Similarly, using approximations (14), (16), and (21), Eqs. (17a)-(17c) can be simplified as follows:

$$\hat{m}_{l,m} = m_{l,m} + O(h^4 + k^2h^2), \quad (22a)$$

$$\hat{m}_{l+1,m} = m_{l+1,m} + O(-h^3 - h^4 - k^2h - k^2h^2), \quad (22b)$$

$$\hat{m}_{l-1,m} = m_{l-1,m} + O(h^3 - h^4 + k^2h - k^2h^2). \quad (22c)$$

Now, from Eqs. (19a)-(19c), we can get:

$$\hat{F}_{l,m} = F_{l,m} + O(h^4 + k^2 + k^2h^2), \quad (23a)$$

$$\hat{F}_{l+1,m} = F_{l+1,m} + O(-h^3 - h^4 + k^2 + k^2h + k^2h^2), \quad (23b)$$

$$\hat{F}_{l-1,m} = F_{l-1,m} + O(h^3 - h^4 + k^2 - k^2h + k^2h^2). \quad (23c)$$

By using Taylor series expansion about grid point (x_l, y_m), Eq. (1) can be written as follows:

$$L_u \equiv \lambda^2(U_{l+1,m} - 2U_{l,m} + U_{l-1,m}) + \frac{k^2}{12}\left[\overline{U}_{yyl+1,m} + \overline{U}_{yyl-1,m} + 10\overline{U}_{yyl,m}\right] = \frac{k^2}{12}\left[\hat{F}_{l+1,m} + \hat{F}_{l-1,m} + 10\hat{F}_{l,m}\right] + \hat{T}_{l,m}, \quad (24)$$

where $l = 1, \cdots, N_x$, $m = 1, \cdots, N_y$, and $\hat{T}_{l,m}$ is the local truncation error. Finally, using the above approximations and from Eq. (24), we can obtain local truncation error, $\hat{T}_{l,m} = O(k^4 + k^4h^2 + k^2h^4)$.

3. Application to singular problem

Consider the two-dimensional elliptic equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = D(x)\frac{\partial u}{\partial x} + g(x,y), \quad 0 < x, \quad y < 1, \quad (25)$$

subject to appropriate Dirichlet boundary conditions prescribed, where functions $D(x)$ and $g(x,y) \in C^2(\Omega)$. Applying Eq. (24) to the above equation, the following difference scheme can be obtained:

$$\lambda^2(U_{l+1,m} - 2U_{l,m} + U_{l-1,m}) + \frac{k^2}{12}\left[\overline{U}_{yyl+1,m} + \overline{U}_{yyl-1,m} + 10\overline{U}_{yyl,m}\right] = \frac{k^2}{12}\left[D_{l+1}\hat{m}_{l+1,m} + D_{l-1}\hat{m}_{l-1,m} + 10D_l\hat{m}_{l,m}\right] + \frac{k^2}{12}\left[g_{l+1,m} + g_{l-1,m} + 10g_{l,m}\right] + \hat{T}_{l,m}. \quad (26)$$

Substituting above approximations (14)-(18) into Eq. (26) will result in:

$$\begin{aligned}
&\{-24\lambda^2-\lambda^2 h(D_{l-1}-D_{l+1})\\
&\quad-2[10+\alpha h(D_{l+1}-D_{l-1})]\\
&\quad+2k^2(\gamma D_{l+1}-5\alpha D_l)D_{l+1}\\
&\quad-2k^2(5\alpha D_l-\gamma D_{l-1})D_{l-1}\}U_{l,m}\\
&\quad+\Big\{12\lambda^2-\lambda^2 h(D_{l+1}+5D_l)\\
&\quad-2[1+h(\gamma D_{l+1}-5\alpha D_l)]\\
&\quad-\frac{3}{2}k^2(\gamma D_{l+1}-5\alpha D_l)D_{l+1}\\
&\quad-\frac{1}{2}k^2(\alpha D_{l+1}-\alpha D_{l-1})D_l\\
&\quad+\frac{1}{2}k^2(5\alpha D_l-\gamma D_{l-1})D_{l-1}\Big\}U_{l+1,m}\\
&\quad+\Big\{12\lambda^2+\lambda^2 h(5D_l+D_{l-1})\\
&\quad-2[1+h(5\alpha D_l-\gamma D_{l-1})]\\
&\quad-\frac{1}{2}k^2(\gamma D_{l+1}-5\alpha D_l)D_{l+1}\\
&\quad+\frac{1}{2}\alpha k^2(D_{l+1}-D_{l-1})D_l\\
&\quad+\frac{3}{2}k^2(5\alpha D_l-\gamma D_{l-1})D_{l-1}\Big\}U_{l-1,m}\\
&\quad+[1+h(\gamma D_{l+1}-5\alpha D_l)](U_{l+1,m+1}+U_{l+1,m-1})\\
&\quad+[10+\alpha h(D_{l+1}-D_{l-1})](U_{l,m+1}+U_{l,m-1})\\
&\quad+[1+h(5\alpha D_l-\gamma D_{l-1})](U_{l-1,m+1}+U_{l-1,m-1})\\
&=k^2\{[1+h(\gamma D_{l+1}-5\alpha D_l)]g_{l+1,m}\\
&\quad+[1+h(5\alpha D_l-\gamma D_{l-1})]g_{l-1,m}\\
&\quad+[10+\alpha h(D_{l+1}-D_{l-1})]g_{l,m}\}+\hat{T}_{l,m}. \qquad (27)
\end{aligned}$$

If the singular terms, like $\frac{1}{x}$, appear in functions $D(x)$ and/or $g(x,y)$, unable to be evaluated at $x=0$, the following approximations are considered:

$$D_{l\pm1}=D_{00}\pm hD_{10}+\frac{h^2}{2}D_{20}\pm O(h^3),$$

$$g_{l\pm1,m}=g_{00}\pm hg_{10}+\frac{h^2}{2}g_{20}\pm O(h^3),$$

where $g_{l,m}=g_{00}=g(x_l,y_m)$, etc. Thus, by neglecting the higher order terms and local truncation error, Eq. (27) can be written as follows:

$$\begin{aligned}
&a_1U_{l,m}+a_2U_{l+1,m}+a_3(U_{l+1,m+1}\\
&\quad+U_{l+1,m-1})+a_4(U_{l,m+1}+U_{l,m-1})\\
&\quad+a_5U_{l-1,m}+a_6(U_{l-1,m+1}+U_{l-1,m-1})\\
&=k^2\{12g_{00}+h^2[g_{20}+2(\gamma-5\alpha)D_{00}g_{10}\\
&\quad+2(\alpha+\gamma)D_{10}g_{00}]\}=G_{l,m}, \qquad (28)
\end{aligned}$$

where:

$$\begin{aligned}
a_1=&-24\lambda^2+2\lambda^2h^2D_{10}-2(10+2\alpha h^2D_{10})\\
&+4k^2(\gamma-5\alpha)D_{00}D_{00},\\
a_2=&12\lambda^2-\lambda^2 h\left(6D_{00}+\frac{h^2}{2}D_{20}\right)\\
&-\lambda^2h^2D_{10}2\left[1+(\gamma-5\alpha)hD_{00}+\gamma h^2D_{10}\right]\\
&-2k^2(\gamma-5\alpha)D_{00}D_{00}-2hk^2(\gamma-2\alpha)D_{00}D_{10},\\
a_3=&1+(\gamma-5\alpha)hD_{00}+\gamma h^2D_{10},\\
a_4=&10+2\alpha h^2D_{10},\\
a_5=&12\lambda^2+\lambda^2 h\left(6D_{00}+\frac{h^2}{2}D_{20}\right)\\
&-\lambda^2h^2D_{10}-2\left[1-(\gamma-5\alpha)hD_{00}+\gamma h^2D_{10}\right]\\
&-2k^2(\gamma-5\alpha)D_{00}D_{00}+2hk^2(\gamma-2\alpha)D_{00}D_{10},\\
a_6=&1-(\gamma-5\alpha)hD_{00}+\gamma h^2D_{10}.
\end{aligned}$$

This modified equation retains its order of accuracy everywhere throughout the solution region, especially in the vicinity of the singularity. Note that this modified scheme Eq. (29) is applicable to both singular and non-singular elliptic equations of form (25).

4. Spline group explicit method

In 1986, Yousif and Evans [6] developed Group Explicit (GE) iterative method, where a small group of 2, 4, 9, 16, and 25 points was constructed in the iterative processes for solving the Laplace's equation. The numerical results showed that the GE method is simpler to program compared to block (line) iterative methods and it requires less storage. However, this method was solely formulated using the usual standard finite-difference discretization which restricts the solutions at only certain points of the solution domain.

Here, we adopt the idea in using non-polynomial spline in the formulation of the group methods. By applying Eq. (29) to any group of four points on the solution domain (as shown in Figure 1), a (4×4) system can be obtained as follows:

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_1 & a_4 & a_6 \\ a_6 & a_4 & a_1 & a_5 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \begin{bmatrix} u_{l,m} \\ u_{l+1,m} \\ u_{l+1,m+1} \\ u_{l,m+1} \end{bmatrix} = \begin{bmatrix} R_{l,m} \\ R_{l+1,m} \\ R_{l+1,m+1} \\ R_{l,m+1} \end{bmatrix}, \quad (29)$$

where:

$$R_{l,m} = - a_5 u_{l-1,m} - a_3 u_{l+1,m-1} - a_6(u_{l-1,m+1} + u_{l-1,m-1}) - a_4 u_{l,m-1} + G_{l,m},$$

$$R_{l+1,m} = - a_2 u_{l+2,m} - a_3(u_{l+2,m+1} + u_{l+2,m-1}) - a_6 u_{l,m-1} - a_4 u_{l+1,m-1} + G_{l+1,m},$$

$$R_{l+1,m+1} = - a_2 u_{l+2,m+1} - a_3(u_{l+2,m+2} + u_{l+2,m}) - a_6 u_{l,m+2} - a_4 u_{l+1,m+2} + G_{l+1,m+1},$$

$$R_{l,m+1} = - a_5 u_{l-1,m+1} - a_3 u_{l+1,m+2} - a_6(u_{l-1,m+2} + u_{l-1,m}) - a_4 u_{l,m+2} + G_{l,m+1}.$$

The (4 × 4) system in Eq. (30) can be inverted and written in explicit forms:

$$\begin{bmatrix} u_{l,m} \\ u_{l+1,m} \\ u_{l+1,m+1} \\ u_{l,m+1} \end{bmatrix} = \frac{1}{\det} \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_5 & b_1 & b_4 & b_6 \\ b_6 & b_4 & b_1 & b_5 \\ b_4 & b_3 & b_2 & b_1 \end{bmatrix} \begin{bmatrix} R_{l,m} \\ R_{l+1,m} \\ R_{l+1,m+1} \\ R_{l,m+1} \end{bmatrix}, \quad (30)$$

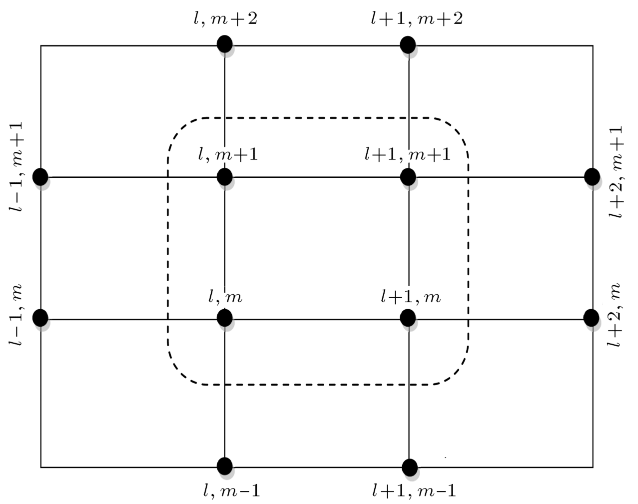


Figure 1. Computational molecule.

where:

$$\det = a_1^4 - 2a_1^2 a_4^2 + a_4^4 - 2a_1^2 a_2 a_5 + 4a_1 a_3 a_4 a_5 - 2a_2 a_4^2 a_5 + a_2^2 a_5^2 - a_3^2 a_5^2 - 2a_1^2 a_3 a_6 + 4a_1 a_2 a_4 a_6 - 2a_3 a_4^2 a_6 - a_2^2 a_6^2 + a_3^2 a_6^2,$$

and:

$$b_1 = a_1^3 - a_1 a_4^2 - a_1 a_2 a_5 + a_3 a_4 a_5 - a_1 a_3 a_6 + a_2 a_4 a_6,$$

$$b_2 = -a_1^2 a_2 + 2a_1 a_3 a_4 - a_2 a_4^2 + a_2^2 a_5 - a_3^2 a_5,$$

$$b_3 = -a_1^2 a_3 + 2a_1 a_2 a_4 - a_3 a_4^2 - a_2^2 a_6 + a_3^2 a_6,$$

$$b_4 = -a_1^2 a_4 + a_4^3 + a_1 a_3 a_5 - a_2 a_4 a_5 + a_1 a_2 a_6 - a_3 a_4 a_6,$$

$$b_5 = -a_1^2 a_5 - a_4^2 a_5 + a_2 a_5^2 + 2a_1 a_4 a_6 - a_2 a_6^2,$$

$$b_6 = 2a_1 a_4 a_5 - a_3 a_5^2 - a_1^2 a_6 - a_4^2 a_6 + a_3 a_6^2.$$

The Gauss-Seidel technique is employed to accelerate the convergence process. Iterations are generated in the groups of four points over the entire spatial domain until the convergence test is satisfied.

Applying System (31) to each of the group in natural row ordering (Figure 2) will lead to a linear system:

$$Au = b,$$

where the matrix of coefficient A is given by:

$$A = \begin{bmatrix} D & U & \\ L & D & U \\ & L & D \end{bmatrix}, \quad (31)$$

with:

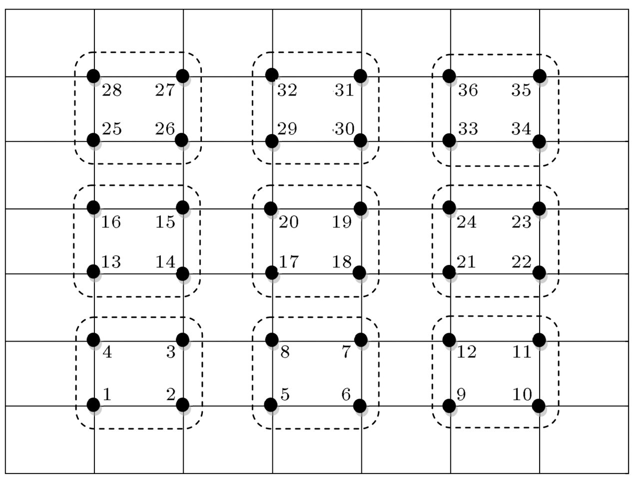


Figure 2. Points ordering for non-polynomial spline group explicit method.

$$D = \begin{bmatrix} R_0 & R_2 & \\ R_5 & R_0 & R_2 \\ & R_5 & R_0 \end{bmatrix}, \qquad U = \begin{bmatrix} R_4 & R_3 & \\ R_6 & R_4 & R_3 \\ & R_6 & R_4 \end{bmatrix},$$

$$L = \begin{bmatrix} R'_4 & R'_3 & \\ R'_6 & R'_4 & R'_3 \\ & R'_6 & R'_4 \end{bmatrix}.$$

The submatrices are given by:

$$R_0 = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_1 & a_4 & a_6 \\ a_6 & a_4 & a_1 & a_5 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a_2 & 0 & 0 & a_3 \\ a_3 & 0 & 0 & a_2 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$R_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_6 & a_4 & 0 & 0 \\ a_4 & a_3 & 0 & 0 \end{bmatrix},$$

$$R_5 = \begin{bmatrix} 0 & a_5 & a_6 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_6 & a_5 & 0 \end{bmatrix}, \qquad R_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_6 & 0 & 0 \end{bmatrix},$$

$$R'_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R'_4 = \begin{bmatrix} 0 & 0 & a_3 & a_4 \\ 0 & 0 & a_4 & a_6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$R'_6 = \begin{bmatrix} 0 & 0 & a_6 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In order to derive the explicit formulae, matrix A is transformed into A^E and vector b is modified into b^E, where:

$$A^E = \text{diag}\{R_0^{-1}\}A,$$

$$b^E = \text{diag}\{R_0^{-1}\}b.$$

The block structure of A^E is the same as that of matrix A with nonzero block R_0 replaced by identity matrices, I and blocks R_i and R'_j, replaced by $R_0^{-1}R_i$, $i = 0, 2, 3, 4, 5, 6$ and $R_0^{-1}R'_j$, $j = 3, 4, 6$, respectively. Since coefficient matrix (32) is block tridiagonal with non-vanishing diagonal element, it is π-consistently ordered and has property-$A^{(\pi)}$ [24]. Thus, the theory of block SOR is also applicable to the non-polynomial spline group explicit iterative method and, therefore, is convergent.

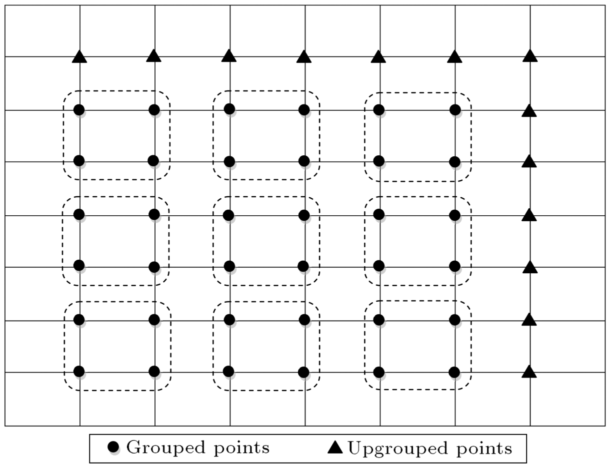


Figure 3. Types of points in non-polynomial spline group explicit for $N_x = N_y = 8$.

Here, the computational complexity of the non-polynomial spline iterative method is examined to show the efficiency of the proposed method. We assume that the solution domain is discretized into even intervals, N_x and N_y in x- and y-directions, respectively. Therefore, we have $(n_x - 1)(n_y - 1)$ Grouped Points (GP) and $(n_x + n_y - 1)$ ungrouped points (UGP), where $n_x = N_x - 1$ and $n_y = N_y - 1$. This can be shown as in Figure 3.

The estimation of this computational complexity is based on the arithmetic operations performed at each iteration for the Additions/Substractions (A/S) and Multiplications/Divisions (M/D) operations [6]. Therefore, the number of operations required per iteration for non-polynomial spline group explicit is given as in Table 1. The total number of arithmetic operations can be obtained by multiplying the number of arithmetic operations for each iteration with the number of iterations.

5. Numerical results

In this section, some benchmark test problems with the known exact solution are solved by the proposed combination of Non-polynomial Spline and Group Explicit iterative method (NSGE), which is approximately

Table 1. The number of arithmetic operations per iteration for non-polynomial spline group explicit iterative method.

	Internal points	**A/S**	**M/D**
GP	$(n_x - 1)(n_y - 1)$	$8(n_x - 1)(n_y - 1)$	$8(n_x - 1)(n_y - 1)$
UGP	$(n_x + n_y - 1)$	$8(n_x + n_y - 1)$	$6(n_x + n_y - 1)$
Total	$n_x n_y$	$8 n_x n_y$	$8 n_x n_y - 2(n_x + n_y - 1)$

$O(k^2+h^4)$. To demonstrate the method is of the fourth order; so, we take $k = h^2$. Relation (12) is suitable for solving Eq. (1), provided that it satisfies the consistency condition; that is, when $\alpha + 2\gamma + \alpha = 1$, which is equivalent to equation $\tan(\omega/2) = \omega/2$. This equation has infinite numbers or roots, the smallest positive nonzero root being given by $\omega = 8.986818916\cdots$ [25].

The results are then compared with those obtained by the:

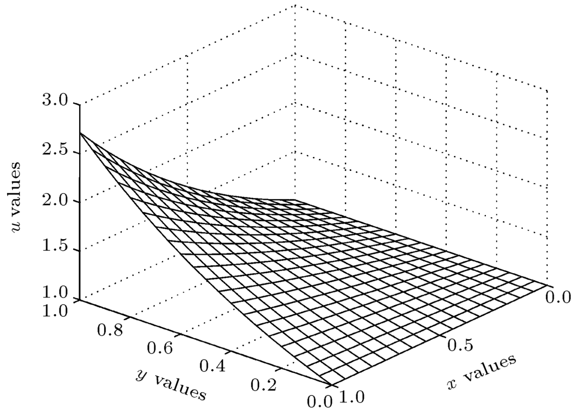
- Combination of non-polynomial spline with standard point Gauss-Seidel iterative method (NSPT);
- Combination of Central Difference scheme with Group Explicit iterative method (CDGE).

where the CDGE scheme is of $O(h^2 + k^2)$, which can be derived by substituting the partial derivative into Eq. (1) with the central difference approximation, similar to the one adopted in [6]. In all cases, we assume that $u^{(0)} = 0$ as the initial guess and the iterations are stopped when the estimated error is below tolerance, that is, when $|u^{(s+1)} - u^{(s)}| \le 10^{-12}$ is achieved. All the experiments are implemented on a PC with Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66 GHz, 3 GB of RAM running Windows 7 using Matlab 7.10.0 (R2010a).

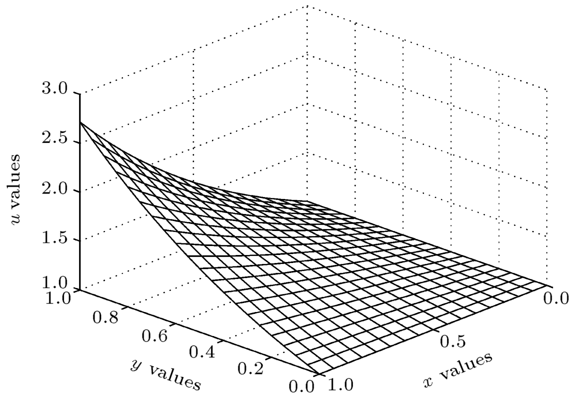
Example 1. Consider the following two-dimensional Poisson's equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{xy}, \qquad 0 < x, \quad y < 1,$$

with Dirichlet boundary conditions satisfying exact solution $u(x,y) = e^{xy}$. The solutions can be obtained by substituting the above approximations (Eqs. (14a)-(14c)), $\hat{m}_{l,m} = \hat{m}_{l+1,m} = \hat{m}_{l-1,m} = 0$ and $g(x,y) = (x^2+y^2)e^{xy}$ into the difference scheme (Eq. (26)) and solved as in Section 4. The graphs of the numerical and exact solutions are plotted in Figure 4(a) and (b), respectively, for $h = 1/16$ and $k = 1/20$. The maximum errors and execution timings of NSGE method compared with those of NSPT method are tabulated in Table 2, while the maximum errors and execution timings obtained by the proposed NSGE method and the existing central difference group scheme CDGE are shown in Table 3. The total arithmetic operations needed for both NSGE and NSPT are displayed in Table 4.



(a) Numerical solution



(b) Exact solution

Figure 4. Simple Poisson's equation for $h = 1/16$ and $k = 1/20$.

Example 2. Consider the convection-diffusion equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \beta \frac{\partial u}{\partial x}, \qquad 0 < x, \quad y < 1,$$

where constant $\beta > 0$ represents the ratio of convection

Table 2. Maximum absolute errors for Example 1 ($k = h^2$).

h	k	$O(k^2+h^4)$-method, NSGE		$O(k^2+h^4)$-method, NSPT	
		Maximum absolute errors	**Time (seconds)**	**Maximum absolute errors**	**Time (seconds)**
$\frac{1}{4}$	$\frac{1}{16}$	0.66375E-05	0.01	0.66375E-05	0.01
$\frac{1}{8}$	$\frac{1}{64}$	0.41322E-06	0.12	0.41317E-06	0.22
$\frac{1}{16}$	$\frac{1}{256}$	0.24756E-07	13.00	0.23968E-07	17.64
$\frac{1}{32}$	$\frac{1}{1024}$	0.23371E-07	1520.14	0.43323E-07	2119.32

Table 3. Maximum absolute errors for Example 1 ($k = h^2$).

h	k	$O(k^2 + h^4)$-method, NSGE		$O(h^2 + k^2)$-method, CDGE	
		Maximum absolute errors	Time (seconds)	Maximum absolute errors	Time (seconds)
$\frac{1}{4}$	$\frac{1}{16}$	0.66375E-05	0.01	0.10415E-03	0.05
$\frac{1}{8}$	$\frac{1}{64}$	0.41322E-06	0.12	0.27036E-04	0.12
$\frac{1}{16}$	$\frac{1}{256}$	0.24756E-07	13.00	0.67558E-05	11.25
$\frac{1}{32}$	$\frac{1}{1024}$	0.23371E-07	1520.14	0.16749E-05	1425.48

Table 4. Total arithmetic operations needed to generate the above results.

h	k	$O(k^2 + h^4)$-method, NSGE		$O(k^2 + h^4)$-method, NSPT	
		Number of iterations	Total arithmetic operations	Number of iterations	Total arithmetic operations
$\frac{1}{4}$	$\frac{1}{16}$	224	153,664	311	167,940
$\frac{1}{8}$	$\frac{1}{64}$	2408	16,658,544	4124	21,824,208
$\frac{1}{16}$	$\frac{1}{256}$	32362	1,963,143,644	56953	2,614,142,700
$\frac{1}{32}$	$\frac{1}{1024}$	445889	225,308,603,478	783787	298,274,845,572

to diffusion and the exact solution is given by:

$$u(x,y)=e^{\frac{\beta x}{2}}\frac{\sin \pi y}{\sinh \sigma}\left[2e^{\frac{-\beta}{2}}\sinh \sigma x+\sinh \sigma(1-x)\right],$$

where $\sigma^2 = \pi^2 + \frac{\beta^2}{4}$. The boundary conditions can be obtained from the exact solution. (4×4) matrix system can be obtained by substituting $D_{00} = \beta$, $D_{10} = D_{20} = 0$ and $G_{l,m} = G_{l+1,m} = G_{l+1,m+1} = G_{l,m+1} = 0$ into Eq. (30). The graphs of the numerical and exact solutions are plotted in Figure 5(a) and (b), respectively, for $\beta = 10$, $h = 1/16$, and $k = 1/20$. Table 5 displays the comparison of maximum errors and execution timings of the NSGE method with those of the NSPT method. Meanwhile, Table 6 depicts the maximum errors and execution timings obtained by the proposed NSGE method compared with those of the existing central difference group scheme CDGE [6]. Table 7 shows the total arithmetic operations needed for both NSGE and NSPT.

Example 3. Given the two-dimensional Poisson's equation in polar cylindrical coordinates in $r - z$ plane:

$$\frac{\partial^2 u}{\partial r^2}+\frac{\partial^2 u}{\partial z^2}+\frac{1}{r}\frac{\partial u}{\partial r}=\cosh z\left(5r\cosh r+2(2+r^2)\sinh r\right),$$

where $0 < r, z < 1$. The exact solution is $u(r,z) = r^2 \sinh r \cosh z$. The solutions can be approximated by replacing variables (x,y) by (r,z) and substituting $g(r,z) = \cosh z(5r\cosh r + 2(2+r^2)\sinh r)$ and $D(r) = -\frac{1}{r}$ into the above scheme (Eq. (29)). The graphs of the numerical and exact solutions are plotted in Figure 6(a) and (b), respectively, for $h = 1/32$, and $k = 1/40$. The maximum errors and execution timings of the proposed NSGE method compared with those of NSPT are displayed in Table 8, while the maximum errors and execution timings of NSGE method compared with those of CDGE are tabulated in Table 9. The total arithmetic operations needed for both NSGE and NSPT are shown in Table 10.

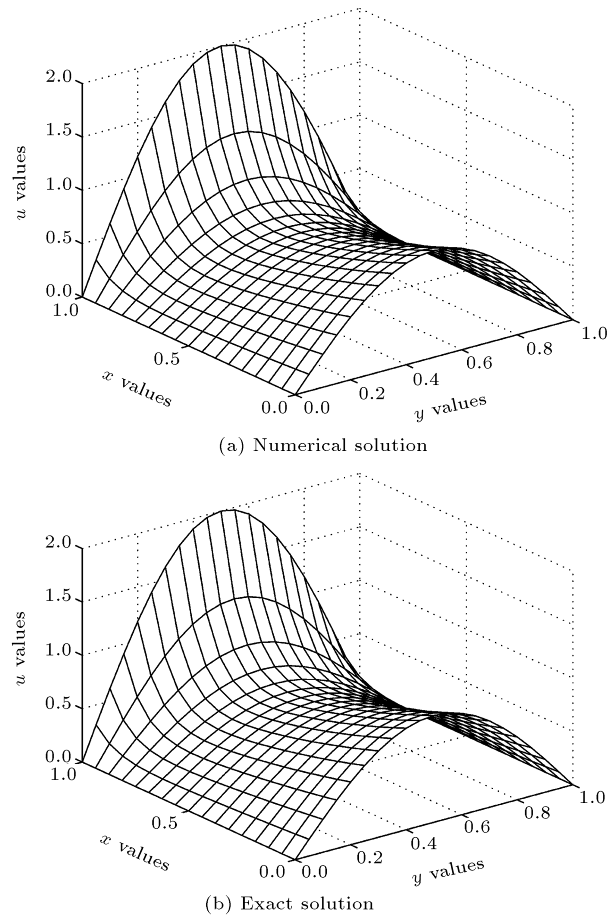


Figure 5. Convection-diffusion equation for $\beta = 10$ and $h = 1/16$.

Table 5. Maximum absolute errors for $k = h^2$, and $\beta = 10$.

h	k	$O(k^2+h^4)$-method, NSGE		$O(k^2+h^4)$-method, NSPT	
		Maximum absolute errors	Time (seconds)	Maximum absolute errors	Time (seconds)
$\frac{1}{4}$	$\frac{1}{16}$	0.35075E-00	0.01	0.35075E-00	0.03
$\frac{1}{8}$	$\frac{1}{64}$	0.21113E-01	0.07	0.21113E-01	0.09
$\frac{1}{16}$	$\frac{1}{256}$	0.10942E-02	5.68	0.10942E-02	8.05
$\frac{1}{32}$	$\frac{1}{1024}$	0.68663E-04	657.93	0.68655E-04	889.61

Table 6. Maximum absolute errors for $k = h^2$, and $\beta = 10$.

h	k	$O(k^2+h^4)$-method, NSGE		$O(h^2+k^2)$-method, CDGE	
		Maximum absolute errors	Time (seconds)	Maximum absolute errors	Time (seconds)
$\frac{1}{4}$	$\frac{1}{16}$	0.35075E-00	0.01	0.24230E-00	0.04
$\frac{1}{8}$	$\frac{1}{64}$	0.21113E-01	0.07	0.78743E-01	0.06
$\frac{1}{16}$	$\frac{1}{256}$	0.10942E-02	5.68	0.17155E-01	5.72
$\frac{1}{32}$	$\frac{1}{1024}$	0.68663E-04	657.93	0.43934E-02	544.28

Table 7. Total arithmetic operations needed to generate the above results.

h	k	$O(k^2+h^4)$-method, NSGE		$O(k^2+h^4)$-method, NSPT	
		Number of iterations	Total arithmetic operations	Number of iterations	Total arithmetic operations
$\frac{1}{4}$	$\frac{1}{16}$	116	74,356	231	135,135
$\frac{1}{8}$	$\frac{1}{64}$	1201	7,778,877	2129	12,205,557
$\frac{1}{16}$	$\frac{1}{256}$	15513	881,712,381	27663	1,375,542,675
$\frac{1}{32}$	$\frac{1}{1024}$	214623	101,643,091,947	380024	156,672,114,456

Table 8. Maximum absolute errors for $k = h^2$.

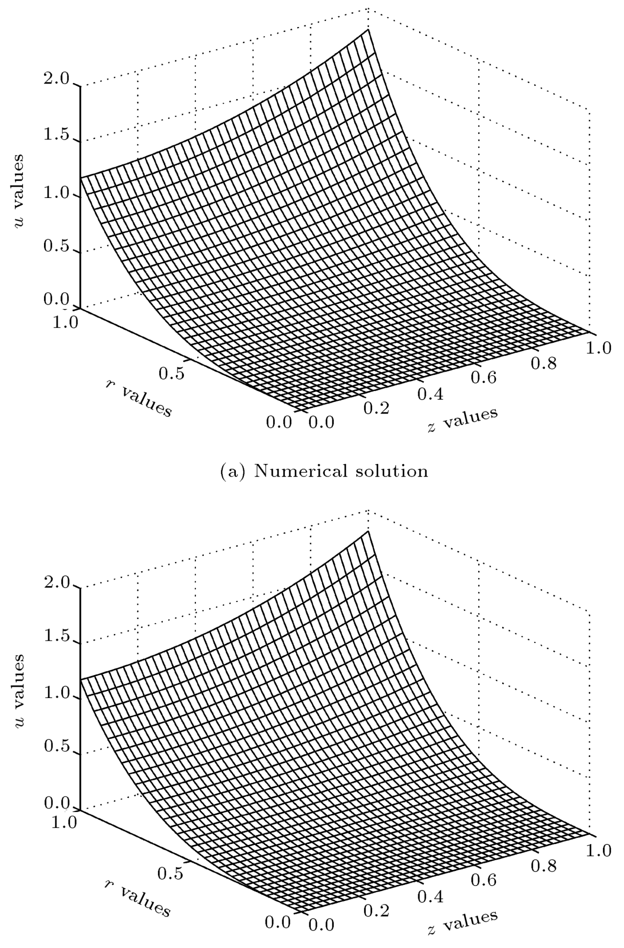
h	k	$O(k^2+h^4)$-method, NSGE		$O(k^2+h^4)$-method, NSPT	
		Maximum absolute errors	Time (seconds)	Maximum absolute errors	Time (seconds)
$\frac{1}{4}$	$\frac{1}{16}$	0.18086E-01	0.04	0.18086E-01	0.04
$\frac{1}{8}$	$\frac{1}{64}$	0.60070E-03	0.21	0.60070E-03	0.23
$\frac{1}{16}$	$\frac{1}{256}$	0.26865E-04	22.51	0.26866E-04	22.92
$\frac{1}{32}$	$\frac{1}{1024}$	0.77407E-05	2544.47	0.77324E-05	2694.46

Table 9. Maximum absolute errors for $k = h^2$.

h	k	$O(k^2+h^4)$-method, NSGE		$O(h^2+k^2)$-method, CDGE	
		Maximum absolute errors	Time (seconds)	Maximum absolute errors	Time (seconds)
$\frac{1}{4}$	$\frac{1}{16}$	0.18086E-01	0.04	0.22704E-01	0.10
$\frac{1}{8}$	$\frac{1}{64}$	0.60070E-03	0.21	0.64553E-02	0.24
$\frac{1}{16}$	$\frac{1}{256}$	0.26865E-04	22.51	0.17345E-02	20.40
$\frac{1}{32}$	$\frac{1}{1024}$	0.77407E-05	2544.47	0.45895E-03	2616.19

Table 10. Total arithmetic operations needed to generate the above results.

h	k	$O(k^2+h^4)$-method, NSGE		$O(k^2+h^4)$-method, NSPT	
		Number of iterations	Total arithmetic operations	Number of iterations	Total arithmetic operations
$\frac{1}{4}$	$\frac{1}{16}$	219	150,234	278	175,140
$\frac{1}{8}$	$\frac{1}{64}$	2494	17,253,492	4183	25,825,842
$\frac{1}{16}$	$\frac{1}{256}$	34002	2,062,629,324	59211	3,170,749,050
$\frac{1}{32}$	$\frac{1}{1024}$	468712	23,6841,111,024	818361	363,337,553,502



(a) Numerical solution

(b) Exact solution

Figure 6. Poisson's equation for $h = 1/32$ and $k = 1/40$.

6. Discussion

It can be observed that for all the model problems, the graph of the numerical solutions almost coincide with that of the exact solutions for different values of x and y, indicating that the computed solutions are in good agreement with the exact ones. As depicted in Tables 2, 5, and 8, the proposed NSGE converges faster than the existing NSPT, which is due to the lower computational complexity of the NSGE method. From Tables 3, 6, and 9, it can be seen that the proposed NSGE produces more accurate results than CDGE which is of $O(h^2+k^2)$, while maintaining almost the same execution timings for all the examples. The total arithmetic operations needed for both NSGE and NSPT, for Examples 1, 2, and 3, are tabulated in Tables 4, 7, and 10. It is clear that the total number of arithmetic operations for NSGE is lower than that of NSPT in all cases due to the grouping strategies in the former method. The gains in the execution timings of NSGE over the NSPT are in the range of 26.3%-45.5% in Example 1, 22.2%-66.7% in Example 2, and 1.8-8.7% in Example 3.

7. Conclusions

In this paper, a new method, which incorporates a non-polynomial spline with the four-point group explicit iterative scheme, was formulated for solving the elliptic boundary value problems. The results show that the proposed method is capable of producing high-accuracy solutions with lesser computation timing compared to the non-polynomial spline standard point Gauss-Seidel method (NSPT) due to its lesser computational complexity. A more accurate result can be obtained by decreasing the step size. However, the computation time (computation cost) will be increased. In addition, the proposed method is superior to the original Central Difference Group Explicit (CDGE) iterative method [6] in terms of accuracy, but with almost similar execution timings. In conclusion, the proposed method is a viable alternative approximation tool to solve the elliptic partial differential equations.

Acknowledgment

The authors gratefully acknowledge the financial support from Universiti Sains Malaysia Research University Grant (1001/JPEND/AUPE002) and FRGS Grant (203/PMATHS/6711321) and thank the reviewers for their valuable suggestions.

References

1. Aarão, J., Bradshaw-Hajek, B.H., Miklavcic, S.J. and Ward, D.A. "The extended-domain-eigenfunction method for solving elliptic boundary value problems

with annular domains", *J. Phys. A: Math. Theor.*, **43**, p. 185202 (2010).

2. He, W.-M. "High effective finite element algorithm for elliptic partial differential equation", *Appl. Math. Comput.*, **187**(2), pp. 1567-1573 (2007).
3. Malavi-Arabshahi, S.M. and Dehghan, M. "Preconditioned techniques for solving large sparse linear systems arising from the discretization of the elliptic partial differential equations", *Appl. Math. Comput.*, **188**(2), pp. 1371-1388 (2007).
4. Wang, Y.-M. and Guo, B.-Y. "Fourth-order compact finite difference method for fourth-order nonlinear elliptic boundary value problems", *J. Comput. Appl. Math.*, **221**(1), pp. 76-97 (2008).
5. Xu, H., Zhang, C. and Barron, R. "A new numerical approach to solve an elliptic equation", *Appl. Math. Comput.*, **171**(1), pp. 1-24 (2005).
6. Yousif, W.S. and Evans, D.J. "Explicit group over-relaxation methods for solving elliptic partial differential equations", *Math. Comput. Simul.*, **28**(6), pp. 453-466 (1986).
7. Abdullah, A.R. "The four point explicit decoupled group (EDG) method: A fast Poisson solver", *Int. J. Comput. Math.*, **38**, pp. 61-70 (1991).
8. Yousif, W.S. and Evans, D.J. "Explicit decoupled group iterative methods and their parallel implementations", *Parallel Alg. Appl.*, **7**, pp. 53-73 (1995).
9. Abdullah, A.R. and Ali, N.H.M. "The comparative study of parallel strategies for the solution of elliptic PDE0s", *Parallel Alg. Appl.*, **10**, pp. 93-103 (1996).
10. Ali, N.H.M. and Lee, S.C. "Group accelerated over relaxation methods on rotated grid", *Appl. Math. Comput.*, **191**, pp. 533-542 (2007).
11. Ng, K.F. and Ali, N.H.M. "Performance analysis of explicit group parallel algorithms for distributed memory multicomputer", *Parallel Comput.*, **34**(6-8), pp. 427-440 (2008).
12. Saeed, A.M. and Ali, N.H.M. "On the convergence of the preconditioned group rotated iterative methods in the solution of elliptic PDEs", *Appl. Math. Inf. Sci.*, **5**(1), pp. 65-73 (2011).
13. Bickley, W.G. "Piecewise cubic interpolation and two-point boundary problems", *Comput. J.*, **11**(2), pp. 206-208 (1968).
14. Albasiny, E.L. and Hoskins, W.D. "Cubic spline solutions to two-point boundary value problems", *Comput. J.*, **12**(2), pp. 151-153 (1969).
15. Fyfe, D.J. "The use of cubic splines in the solution of two-point boundary value problems", *Comput. J.*, **12**(2), pp. 188-192 (1969).
16. Al-Said, E.A., Noor, M.A. and Rassias, T.M. "Cubic splines method for solving fourth-order obstacle problems", *Appl. Math. Comput.*, **174**(1), pp. 180-187 (2006).
17. Goh, J., Majid, A.A. and Ismail, A.I.M. "Numerical method using cubic B-spline for the heat and wave equation", *Comput. Math. Appl.*, **62**(12), pp. 4492-4498 (2011).
18. Goh, J., Majid, A.A. and Ismail, A.I.M. "A quartic B-spline for second-order singular boundary value problems", *Comput. Math. Appl.*, **64**(2), pp. 115-120 (2012).
19. Mohanty, R.K. and Gopal, V. "High accuracy cubic spline finite difference approximation for the solution of one-space dimensional non-linear wave equations", *Appl. Math. Comput.*, **218**(8), pp. 4234-4244 (2011).
20. Ding, H.F., Zhang, Y.X., Cao, J.X. and Tian, J.H. "A class of difference scheme for solving telegraph equation by new non-polynomial spline methods", *Appl. Math. Comput.*, **218**(9), pp. 4671-4683 (2012).
21. Gopal, V., Mohanty, R.K. and Jha, N. "New nonpolynomial spline in compression method of $O(k^2+h^4)$ for the solution of 1D wave equation in polar coordinates", *Adv. Numer. Anal.*, **2013**, 8 pages (2013).
22. Jha, N. and Mohanty, R.K. "Quintic hyperbolic non-polynomial spline and finite difference method for nonlinear second order differential equations and its application", *J. Egyptian Math. Soc.*, **22**(1), pp. 115-122 (2014).
23. Goh, J. and Ali, N.H.M. "High accuracy Spline Explicit Group (SEG) approximation for two dimensional elliptic boundary value problems", *Plos One*, **10**(7), pp. 1-13 (2015).
24. Young, D.M., *Iterative Solution of Large Linear Systems*, pp. 140-171, Academic Press, New York (1971).
25. Jain, M.K. and Aziz, T. "Spline function approximation for differential equations", *Comput. Methods in Appl. Mech. Eng.*, **26**(2), pp. 129-143 (1981).

Biographies

Joan Goh received her Bachelor of Science (Maths) with a minor in Management from Universiti Sains Malaysia in 2003. She received her MSc in Mathematics and completed her PhD in Numerical Analysis from Universiti Sains Malaysia, in 2007 and 2013, respectively. She continued her research at Universiti Sains Malaysia as a Postdoctoral Fellow in 2014.

Norhashidah Hj Mohd Ali received her Bachelor of Science (Maths) with a minor in Information Sciences from Western Illinois University, USA. She received her MSc in Applied Mathematics from Virginia Tech, USA, and completed her PhD in Industrial Computing from Universiti Kebangsaan Malaysia in 1998. She has been with Universiti Sains Malaysia since 1990. Her research interests are mainly in the area of Numerical Partial Differential Equations.