Fourth-order numerical solutions of diffusion equation by using SOR method with Crank-Nicolson approach

F A Muhiddin\textsuperscript{1} and J Sulaiman\textsuperscript{2}

\textsuperscript{1}Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Sabah, Malaysia.
\textsuperscript{2}Mathematics with Economics Programme, Universiti Malaysia Sabah, Malaysia.

E-mail: fatih170@sabah.uitm.edu.my, jumat@ums.edu.my

Abstract. The aim of this paper is to investigate the effectiveness of the Successive Over-Relaxation (SOR) iterative method by using the fourth-order Crank-Nicolson (CN) discretization scheme to derive a five-point Crank-Nicolson approximation equation in order to solve diffusion equation. From this approximation equation, clearly, it can be shown that corresponding system of five-point approximation equations can be generated and then solved iteratively. In order to access the performance results of the proposed iterative method with the fourth-order CN scheme, another point iterative method which is Gauss-Seidel (GS), also presented as a reference method. Finally the numerical results obtained from the use of the fourth-order CN discretization scheme, it can be pointed out that the SOR iterative method is superior in terms of number of iterations, execution time, and maximum absolute error.

1. Introduction
Numerous complex models in the fields of engineering, finance, and sciences required numerical approaches in order to be solvable. The partial differential equations (PDEs) are discretized to form an approximation equation which will lead to a large and sparse system of linear equations (SLE). Then this linear system can be solved by means of some iterative methods.

The main concern on the traditional numerical schemes is the difficulties to produce highly accurate numerical approximations and fine discretization is needed [1]. Henceforth, many prior studies on higher-order finite difference schemes can be found in the recent literatures. Some of the methods are fourth-order arithmetic mean scheme [2], new higher-order accurate compact scheme [3], fourth-order compact alternating direction implicit (ADI) method [4], and higher-order compact and Crank-Nicolson ADI scheme [5].

Apart from the discretization approaches, basic iterative methods such as the Richardson, Jacobi and Gauss Seidel (GS) still suffers from their slow convergence rate. Attributes to the advantages of the SOR iterative method, this paper investigates the effectiveness of SOR iterative method in solving one-dimensional (1D) diffusion problems via the fourth-order Crank-Nicolson (CN) finite difference approximation equation. To do this, let the general form of 1D diffusion equation be defined as

\[
\frac{\partial U}{\partial t} = D \frac{\partial^2 U}{\partial x^2}, \quad x \in [a, b], \quad t \in [0, T]
\]

where $D$ is the diffusion parameter.
Equation (1) is a general version of heat equation that satisfies the parabolic partial differential equation. To get an approximate solution of equation (1), we assume that the solution domain in space \( \Delta x \) can be uniformly divided into \((n+1)\) subintervals, while the time step \( \Delta t \) can be divided into uniform \( M \) subintervals.

\[
\Delta x = \frac{(b-a)}{m} = h, \quad m = n + 1, \quad \Delta t = \frac{(T-0)}{M} \tag{2}
\]

2. Fourth-order implicit finite difference approximation

Figure 1 shows the finite grid network that is used to formulate the approximate equation, where the GS and SOR iterative methods are applied onto each interior node point until the convergence test can be reached.

**Figure 1.** The distribution of uniform node points for the solution domain at \( m=8 \).

By using the second-order CN discretization scheme, the general second-order approximation equation for equation (1) can be shown as

\[
-\beta_0 U_{i-1,j} + (1 + 2\beta_0)U_{i,j} - \beta_0 U_{i+1,j} = F_{i,j} \tag{3}
\]

where

\[
\beta_0 = \frac{\Delta t}{2\Delta x^2}, \quad \text{and} \quad F_{i,j} = \beta_0 U_{i-1,j} + (1 - 2\beta_0)U_{i,j} + \beta_0 U_{i+1,j}.
\]

Beside the second-order scheme, the fourth-order CN approximation equation for equation (1) can be stated as

\[
\beta_1 U_{i-2,j} + 16\beta_1 U_{i-1,j} + (1 + 30\beta_1)U_{i,j} - 16\beta_1 U_{i+1,j} + \beta_1 U_{i+2,j} = F_{i,j} \tag{4}
\]

where

\[
\beta_1 = \frac{\Delta t}{24\Delta x^2}, \quad \text{and} \quad F_{i,j} = -\beta_1 U_{i-2,j} + 16\beta_1 U_{i-1,j} + (1 - 30\beta_1)U_{i,j} + 16\beta_1 U_{i+1,j} - \beta_1 U_{i+2,j}.
\]

Then the computational molecule for equations (3) and (4) are shown as in figures 2 and 3 respectively.

**Figure 2.** The computational molecule for second-order CN approximation equations.

**Figure 3.** The computational molecule for fourth-order CN approximation equations.
Based on Figures 1 and 3, apparently the approximate values for two node points, \( i = 1 \) and \( i = m - 1 \) cannot be computed using equation (4), because some of the node points are outside of the solution domain in equation (1). To overcome this problem, the corresponding second-order approximation equation (3) needs to be applied together at two interior node points, \( i = 1 \) and \( i = m - 1 \). Therefore, the combination of equations (3) and (4) can easily lead to a linear system in matrix form as

\[
AU = F
\]  

where

\[
A = \begin{bmatrix}
\alpha_0 & -\beta_0 & & & \\
-16\beta_1 & \alpha_1 & -16\beta_1 & \beta_1 & \\
& -16\beta_1 & \alpha_1 & -16\beta_1 & \beta_1 & \\
& & \ddots & \ddots & \ddots & \ddots & \\
& & \beta_1 & -16\beta_1 & \alpha_1 & -16\beta_1 & \beta_1 \\
& & & \beta_1 & -16\beta_1 & \alpha_1 & -16\beta_1 & -\beta_0 & \alpha_0
\end{bmatrix}_{(m-1)\times(m-1)}
\]

\[
U = \begin{bmatrix}
U_{1,j+1} & U_{2,j+1} & \cdots & U_{m-2,j+1} & U_{m-1,j+1}
\end{bmatrix}^T,
\]

\[
F = \begin{bmatrix}
f_{1,j} + \beta_0 U_{0,j+1} & F_{2,j} - \beta_1 U_{0,j+1} & \cdots & F_{m-2,j} - \beta_1 U_{1,j+1} & f_{m-1,j} + \beta_0 U_{m,j+1}
\end{bmatrix}^T,
\]

and

\[
\alpha_0 = 1 + 2\beta_0, \quad \alpha_1 = 1 + 30\beta_1.
\]

3. Derivation of SOR iterative algorithm

In order to solve the sparse and large linear system in equation (5), let the coefficient matrix \( A \) be decomposed as

\[
A = D + L + V
\]

where \( D, L \) and \( V \) are diagonal, lower triangular, and upper triangular matrices respectively. By using the decomposition in equation (6), the general scheme of the SOR method as in Young [6, 7] and Saad [8] can be stated as follows

\[
U_i^{(k+1)} = (1 - \omega)U_i^{(k)} + \omega(D + L)^{-1}(-VU_i^{(k)} + F)
\]

(7)

where, \( \omega \) and \( U_i^{(k)} \) represent the relaxation factor and the unknown vector at the \( k \)th iteration respectively. By taking \( \omega = 1 \), the SOR iterative method can be reduced to GS iterative method. Therefore, the optimal value of \( \omega \) is usually determined within the range of \( 1 \leq \omega < 2 \). Based on the optimal value of \( \omega \), theoretically, the SOR iterative method can be converging faster than the GS iterative method.

In this paper, the GS iterative method is assigned to be used as a control method. The implementation of SOR iterative method to solve diffusion equation can be described in Algorithm 1.

---

**Algorithm 1: SOR scheme**

i. Initialize \( U_i^{(0)} \leftarrow 0 \) and \( \varepsilon \leftarrow 10^{-10} \)

ii. Assign the optimal value of \( \omega \).

iii. For \( i = 1, 2, 3, \ldots, m - 1 \), calculate:

\[
U_i^{(k+1)} = (1 - \omega)U_i^{(k)} + \omega(D + L)^{-1}(-VU_i^{(k)} + F)
\]
iv. Perform the convergence test, \[ \left| U_i^{(k+1)} - U_i^k \right| \leq \epsilon = 10^{-10} \]. If yes, go to step (v). Otherwise repeat step (iii).

v. Display approximate solutions.

4. Numerical experiments
In order to verify the effectiveness of the SOR method together with the fourth-order CN scheme as compared with the GS method, two examples of one-dimensional diffusion equations were tested. For comparison purposes, three parameters are considered, namely the number of iterations \((k)\), execution time \((t)\) which is measured in seconds, and maximum absolute error. Throughout the numerical experiments, the tolerance error of convergence is \( \epsilon = 10^{-10} \).

**Example 1** [9]
By considering \( D=1 \) in equation (1) gives
\[
\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad x \in [0,1], \quad 0 \leq t \leq 0.5
\]
subject to the initial condition, \( U(x,0) = \sin(\pi x) \) and satisfy the exact solution, \( U(x,0) = \sin(\pi x) \exp(-\pi^2 t) \).

**Example 2** [10]
By considering \( D=1 \) in equation (1) gives
\[
\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad x \in [0,1], \quad 0 \leq t \leq 1.0
\]
The initial and boundary conditions and exact solution of the equation (9) are given by:
\[
U(x,t) = e^{-(x^2)} \sin(\pi x) + 3e^{-i(x^2)} \sin(2\pi x), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1.0
\]

Then, the results of numerical experiment for equations (8) and (9) obtained from the implementation of SOR, and GS iterative methods are recorded in Table 1.

| M     | Method | Example 1 | Example 2 |
|-------|--------|-----------|-----------|
|       |        | \( k \)   | \( t \)   | \( \text{Max. Error} \) | \( k \)   | \( t \)   | \( \text{Max. Error} \) |
| 128   | GS     | 716       | 0.41      | 7.2999e-06 | 841       | 0.64      | 5.1532e-07  |
|       | SOR    | 110       | 0.07      | 7.2009e-06 | 103       | 0.08      | 4.1300e-07  |
| 256   | GS     | 2564      | 2.94      | 7.6093e-06 | 2804      | 4.79      | 8.2683e-07  |
|       | SOR    | 210       | 0.26      | 7.2009e-06 | 191       | 0.29      | 4.1277e-07  |
| 512   | GS     | 9125      | 22.26     | 8.8459e-06 | 9005      | 31.95     | 2.0719e-06  |
|       | SOR    | 401       | 1.00      | 7.2011e-06 | 356       | 1.09      | 4.1261e-07  |
| 1024  | GS     | 32045     | 153.02    | 1.3791e-05 | 26758     | 234.32    | 7.0520e-06  |
|       | SOR    | 765       | 3.70      | 7.2022e-06 | 663       | 4.29      | 4.1292e-07  |
| 2048  | GS     | 110396    | 1084.54   | 3.3569e-05 | 59695     | 1587.24   | 2.6972e-05  |
|       | SOR    | 1458      | 14.22     | 7.2063e-06 | 1223      | 16.58     | 4.1331e-07  |
Table 2. Decrement Percentage of the Number of Iterations and the Execution Time for SOR Method Compared to GS Method.

| M   | Number of Iterations (%) | Execution Time (%) |
|-----|--------------------------|--------------------|
|     | Example 1 | Example 2 | Example 1 | Example 2 |
| 128 | 84.64     | 87.75      | 82.93     | 87.50      |
| 256 | 91.81     | 93.19      | 91.16     | 93.95      |
| 512 | 95.61     | 96.05      | 95.51     | 96.59      |
| 1024| 97.61     | 97.52      | 97.5      |            |
| 2048| 98.68     | 97.95      | 98.69     | 98.96      |

5. Conclusion

In this paper, the formulations of SOR and GS iterative methods were derived using the corresponding second- and fourth-order Crank-Nicolson finite difference schemes as shown in equations (3) and (4) respectively. The numerical results have been demonstrated by using two examples. As recorded in Table 1, it clearly shows that by applying SOR iterative method, the number of iterations and execution time of the iterative method can be reduced approximately from 84.64–98.68% and 82.93–98.96% respectively as compared to the GS iterative method. Meanwhile, both numerical examples also show substantial improvement in terms of maximum error value as the mesh size increased with the application of the SOR iterative method. Thus, it can be concluded that the SOR iterative method is indeed superior than the GS iterative method. For future work, this study will be extended to investigate on the use of Modified SOR as a smoother to the proposed approximation equation.

References

[1] Ma Y and Ge Y 2010 Appl Math Comput 215 pp 3408–417
[2] Sulaiman J Othman M and Hasan M K 2005 Proc 1st IMT–GT Reg Conf on Mathematics, Statistics and Their Applications Parapat Indonesia pp 139–46
[3] Han F and Dai W 2013 App Math Model 37 pp 7940–952
[4] Hendricks C, Ehrhardt M and Günther M 2016 Appl Numer Math 101 pp 36–52
[5] Zhang Q, Zhang C and Wang L 2016 J Comput Appl Math 306 pp 217–30
[6] Young D M 2014 Iterative solution of large linear systems (Florida: Academic Press)
[7] Young D M 1972 J Approx Theory 5 pp 37–148
[8] Saad Y 2003 Iterative methods for sparse linear systems 2nd ed (Philadelphia: SIAM)
[9] Sulaiman J, Othman M and Hasan M K 2009 Proc 2nd Int Conf on Mathematical Modelling and Computation (MMC09) and 5th East Asia Industrial and Applied Mathematics Conference (EASIAM 5) Brunei 8–11 June 2009
[10] Sulaiman J, Othman M and Hasan M K 2008 Modeling Simulation and Optimization of Complex Processes (Berlin: Springer–Verlag) pp 547–56
[11] Morton K W and Mayers D F 2005 Numerical solution of partial differential equations: an introduction (2nd ed) (Cambridge: Cambridge University Press)