

## Investigation of Steady State Problems via Quarter Sweep Schemes (Kajian Terhadap Masalah Berkeadaan Mantap Menggunakan Skema Sapuan Suku)

NG YIT HOE\* & MOHAMMAD KHATIM HASAN

### ABSTRACT

*Numerical application helps researchers in simulating various problems and used for solving partial differential equation. Half sweep and quarter sweep approach have been applied onto iterative method to gain approximation solution. In this paper, the implementation of full sweep successive over relaxation (FSSOR), half sweep successive over relaxation (HSSOR) and quarter sweep successive over relaxation (QSSOR) methods and full sweep accelerated over relaxation (FSAOR), half sweep accelerated over relaxation (HSAOR) and quarter sweep accelerated over relaxation (QSAOR) for its numerical engines are shown. QSSOR and QSAOR method was the fastest among FSSOR, HSSOR, FSAOR and HSAOR methods. Additionally, QSAOR performance is more accurate than QSSOR.*

*Keywords: Elliptic problem; iterative scheme; numerical simulation; Poisson; quarter sweep approach*

### ABSTRAK

*Aplikasi berangka membantu penyelidik dalam mensimulasikan pelbagai masalah dan digunakan untuk menyelesaikan persamaan terbitan separa. Pendekatan sapuan separuh dan sapuan suku telah digunakan ke atas kaedah lalaran untuk mendapatkan penyelesaian penghampiran. Dalam kertas ini, implementasi kaedah-kaedah pengenduran berlebihan berturut-turut sapuan penuh (FSSOR), pengenduran berlebihan berturut-turut sapuan separuh (HSSOR) dan pengenduran berlebihan berturut-turut sapuan suku (QSSOR) dan pemecutan berlebihan berturut-turut sapuan penuh (FSAOR), pemecutan berlebihan berturut-turut sapuan separuh (HSAOR) dan pemecutan berlebihan berturut-turut sapuan suku (QSAOR) untuk enjin berangka ditunjukkan. Kaedah QSSOR dan QSAOR adalah terpacat dalam kalangan kaedah-kaedah FSSOR, HSSOR, FSAOR dan HSAOR. Di samping itu, prestasi QSAOR adalah lebih tepat daripada QSSOR.*

*Kata kunci: Masalah eliptik; pendekatan sapuan suku; poisson; simulasi berangka; skim lalaran*

### INTRODUCTION

Finite difference, finite volume, finite element and boundary element have essential roles in simulating scientific problem. Successive over relaxation (SOR) iterative method was first introduced by Young (1950) in solving partial difference equations (PDEs) of elliptic type. Later, Hadjidimos (1978) extended the development of SOR method which replaced SOR method by accelerated over relaxation (AOR) method. The other modified iterative were also proposed by the researchers methods (Ali & Sam 2006; Moussavi 2009; Wang et al. 2011).

The computational complexity of methods could be reduced by applying the half sweep (HS) and quarter sweep (QS) techniques onto the various methods in simulating problems. Sulaiman et al. (2008a) proposed a new half sweep AMG algorithm to solve the diffusion equations for two-point boundary problems. The approach has succeeded in reducing the computational complexity for linear system which involved in one AMG cycle. Sulaiman et al. (2008b) also applied the quarter sweep (QS) approach onto arithmetic means to solve water quality model.

In addition, the complexity reduction approach concepts for different strategies such as lexicography (NA), red black (RB) and four colour (4C) have also achieved

success in reducing the computation time. Sulaiman et al. (2006) implemented the red black (RB) strategy and optimal ordering strategy of the point iterative algorithm to solve the 2D convection-diffusion problem. The authors showed that QSSOR-RB is superior to FSSOR (-NA and -4C), HSSOR (-NA and -4C), QSSOR (-NA and -4C), FSSOR-RB and HSSOR-RB in terms of iterations number and execution time. Sulaiman et al. (2007) implemented the RB strategy to other iterative methods based on the good performance yielded by the RB strategy. In this paper, the authors used three AOR and three SOR schemes for simulation in solving various problems for the steady state phenomenon.

### THEORY AND CONCEPT

A two dimensional Poisson's equation has been used (Lee 2007; Othman et al. 2009) and defined as:

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega. \quad (1)$$

Dirichlet boundary condition derived as in (2) and  $\Omega$  is a continuous unit square solution domain,

$$U(x, y) = g(x, y), \quad (x, y) \in \partial\Omega. \quad (2)$$

A point or a group of iterative method discharged at Poisson's equation will appears as the linear system of equations  $A\underline{u} = \underline{b}$ . The matrix  $A$  is given in (3) with diagonal matrix,  $D$ , lower triangular matrix,  $L$  and upper triangular matrix,  $U$ , respectively,

$$A = D - L - U. \tag{3}$$

AOR method following Hadjidimos (1978) is defined as (4):

$$\begin{aligned} u^{(k+1)} &= Lr_w u^{(k)} + w(D - rL)^{-1}b, \\ L_{r,w} &= (I - rD^{-1} - L)^{-1}[1 - w]I \\ &\quad + (w - r)L + wD^{-1}U. \end{aligned} \tag{4}$$

This method engaged two accelerated parameters which are  $r$  and  $w$ , while  $k$  represents iteration number. Thus, (4) can be rewritten as,

$$(D - rL)\underline{u}^{(k+1)} = (w - r)L\underline{u}^{(k)} + wU\underline{u}^{(k)} + w\underline{b} + (1 - w)D\underline{u}^{(k)}. \tag{5}$$

and

$$D\underline{u}^{(k+1)} = rL(\underline{u}^{(k+1)} - \underline{u}^{(k)}) + wL\underline{u}^{(k)} + wU\underline{u}^{(k)} + w\underline{b} + (1 - w)D\underline{u}^{(k)}. \tag{6}$$

Therefore, the SOR method will be accomplished as in (7),

$$D\underline{u}^{(k+1)} = wL\underline{u}^{(k+1)} + wU\underline{u}^{(k)} + w\underline{b} + (1 - w)D\underline{u}^{(k)}. \tag{7}$$

By substituting  $wL\underline{u}^{(k+1)}$  in (7) with  $wL\underline{u}^{(k)}$  and adding  $rL(\underline{u}^{(k+1)} - \underline{u}^{(k)})$ , the AOR method will be accomplished as in (8).

$$D\underline{u}^{(k+1)} = rL(\underline{u}^{(k+1)} - \underline{u}^{(k)}) + wL\underline{u}^{(k)} + wU\underline{u}^{(k)} + w\underline{b} + (1 - w)D\underline{u}^{(k)}. \tag{8}$$

Finite difference method has been used in solving various problems. Equation (1) can be projected at point  $(x, y)$  with a grid in surface  $(x, y)$  which having lattice distance  $h$  and  $g$ . The directions are  $x_i = ih$  and  $y_j = jg$  (where  $i, j = 0, 1, 2, 3, \dots, N$ ),  $u_{i,j} = u(x,y)$  and  $h = \frac{x}{N}$ ,  $g = \frac{y}{N}$ ,  $N$  is the number of grid point in the axis. In this paper, authors use lattice distance,  $h$  not necessary equal to  $g$ . The computational molecule standard 5 points finite difference scheme has been revealed in Figure 1 (Ali & Evans 2004).

From the 2<sup>nd</sup> order scheme, FSSOR method and QSSOR method generally are written as

$$\begin{aligned} (p)^2(h)^2(g)^2f_{i,j} &= (g)^2u_{i-p,j} + (g)^2u_{i+p,j} \\ &\quad + (h)^2u_{i,j-p} + (h)^2u_{i,j+p} + (h)^2u_{i,j+p} \\ &\quad - 2(h)^2u_{i,j} - 2(g)^2u_{i,j}. \end{aligned} \tag{9}$$

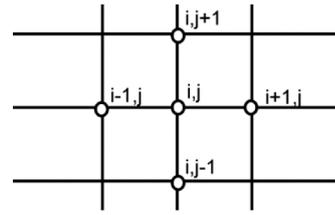


FIGURE 1. Standard 5 Point finite difference scheme

Here, full sweep and quarter sweep cases are  $p = 1$  and  $p = 2$ , respectively. Therefore, the FSSOR and QSSOR schemes are defined as:

$$\begin{aligned} u_{i,j}^{(k+1)} &= w \left( \frac{(g)^2 u_{i-p,j}^{(k+1)} + (g)^2 u_{i+p,j}^{(k)} + h^2 u_{i,j-p}^{(k+1)} + (h)^2 u_{i,j+p}^{(k)} - (p)^2 (g)^2 (h)^2 f_{i,j}}{2((h)^2 + (g)^2)} \right) \\ &\quad + (1-w)u_{i,j}^{(k)}. \end{aligned} \tag{10}$$

By replacing  $u_{i-p,j}^{(k+1)}$  and  $u_{i-j,p}^{(k+1)}$  with  $u_{i-p,j}^{(k)}$  and  $u_{i,j-p}^{(k)}$ , respectively, and adding  $\frac{r((g)^2 u_{i-p,j}^{(k+1)} - (g)^2 u_{i-p,j}^{(k)})}{2((h)^2 + (g)^2)}$  and  $\frac{r((h)^2 u_{i-j,p}^{(k+1)} - (h)^2 u_{i-j,p}^{(k)})}{2((h)^2 + (g)^2)}$ , the FSAOR and QSAOR schemes in (11) can be derived from (4).

$$\begin{aligned} u_{i,j}^{(k+1)} &= r \left( \frac{(g)^2 u_{i-p,j}^{(k+1)} - (g)^2 u_{i-p,j}^{(k)} + (h)^2 u_{i,j-p}^{(k+1)} - (h)^2 u_{i,j-p}^{(k)}}{2((h)^2 + (g)^2)} \right) \\ &\quad + w \left( \frac{(g)^2 u_{i-p,j}^{(k)} + (g)^2 u_{i+p,j}^{(k)} + (h)^2 u_{i,j-p}^{(k)} + (h)^2 u_{i,j+p}^{(k)} - (p)^2 (g)^2 (h)^2 f_{i,j}}{2((h)^2 + (g)^2)} \right) \\ &\quad + (1-w)u_{i,j}^{(k)}. \end{aligned} \tag{11}$$

Rotated five-point finite difference scheme in Figure 2 (Ali & Evans 2004) can be stated as HSSOR method namely:

$$\begin{aligned} ((h)^2 + (g)^2)f_{i,j} &= u_{i-1,j+1} + u_{i+1,j+1} + u_{i+1,j-1} \\ &\quad + u_{i-1,j-1} - 4u_{i,j}. \end{aligned} \tag{12}$$

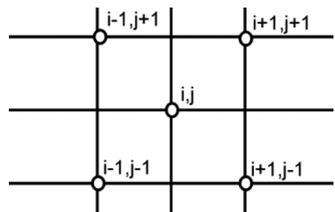


FIGURE 2. Rotated five-point finite difference scheme

Then, HSSOR method as in (12) is discovered by:

$$u_{i,j}^{(k+1)} = w \left( \frac{u_{i-2,j+1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} - ((h)^2 + (g)^2) f_{i,j}}{4} \right) + (1-w)u_{i,j}^{(k)} \tag{13}$$

By applying the same approach as in (4) for HSSOR method, HSAOR method in (14) can be determined by replacing  $u_{i-1,j-1}^{(k+1)}$  and  $u_{i+1,j-1}^{(k+1)}$  with  $u_{i-1,j-1}^{(k)}$  and  $u_{i+1,j-1}^{(k)}$  respectively and adding  $\frac{r(u_{i-1,j-1}^{(k+1)} - u_{i-1,j-1}^{(k)})}{4}$  and  $\frac{r(u_{i+1,j-1}^{(k+1)} - u_{i+1,j-1}^{(k)})}{4}$ .

$$u_{i,j}^{(k+1)} = r \left( \frac{u_{i-1,j-1}^{(k+1)} - u_{i+1,j-1}^{(k+1)} + u_{i-1,j-1}^{(k)} - u_{i+1,j-1}^{(k)}}{4} \right) + w \left( \frac{u_{i-2,j+1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} - (h^2 + g^2) f_{i,j}}{4} \right) + (1-w)u_{i,j}^{(k)} \tag{14}$$

The stopping criteria for the iteration was the maximum test with tolerance:

$$\text{maks} \left\{ |u_{i,j}^{(k+1)} - u_{i,j}^{(k)}| \right\} < 10^{-10} \tag{15}$$

NUMERICAL EXPERIMENT

In this paper, FSSOR, HSSOR, QSSOR, FSAOR, HSAOR and QSAOR iterative methods were developed by using C# and Microsoft visual studio 2008 in order to solve elliptic equations. The performance of these numerical engines with different number of grid points ( $n=N \times N$ ) in all experiments were indicated in Figure 3-12. The results of those iterative methods were exported to Microsoft Excel for further analysis. Obviously, there is no exact solution for experiment 3 and 4. In the experiment, the iterative method with the initial value,  $u_{i,j}^{(0)} = 0$ . Convergence test rate used in this paper is  $\text{maks} \left\{ |u_{i,j}^{(k+1)} - u_{i,j}^{(k)}| \right\} < 10^{-10}$ .

NUMERICAL EXPERIMENT 1 (EXPERIMENT 1)

In experiment 1, the 2<sup>nd</sup> order elliptic equation is considered as:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) = -(kos(x + y) + kos(x - y)), \tag{16}$$

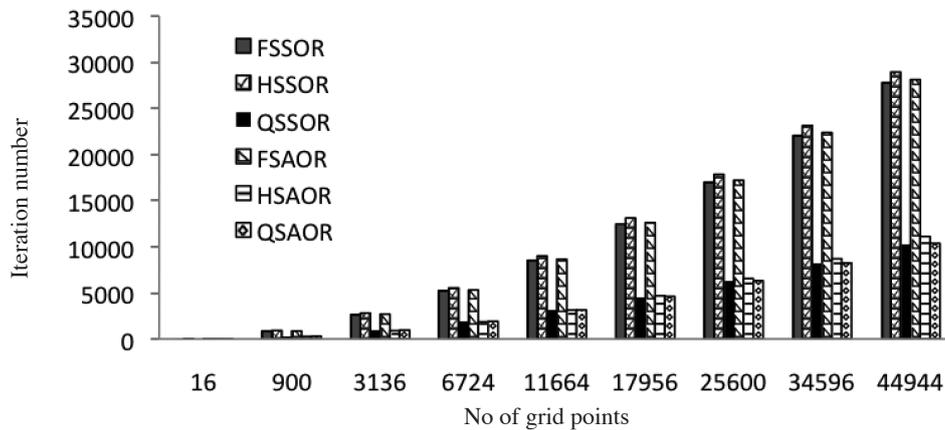


FIGURE 3. Comparison of iteration number for Experiment 1

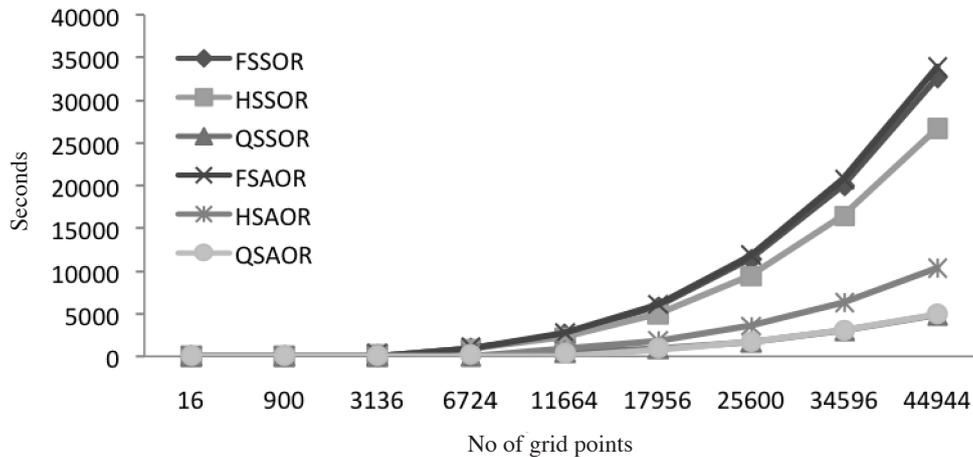


FIGURE 4. Comparison of execution time for Experiment 1

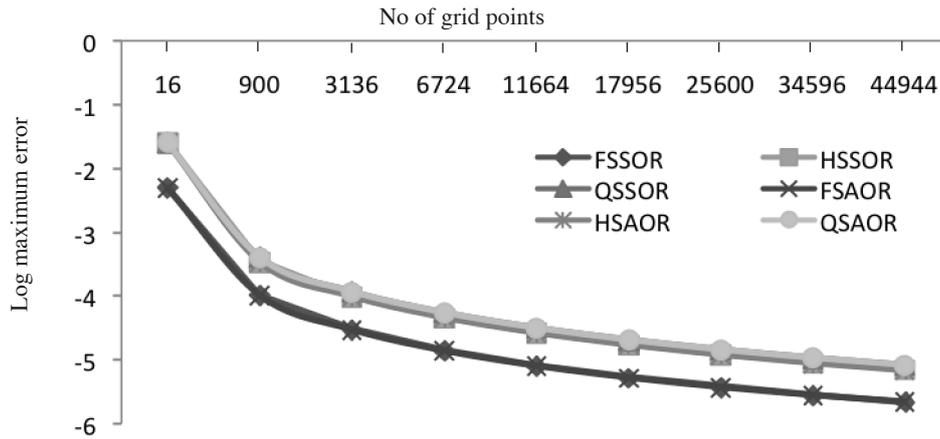


FIGURE 5. Comparison of accuracy for Experiment 1

with region  $R, (x, y) \in (0, \pi) \times (0, \frac{\pi}{2})$ , boundary condition is given by:

$$\begin{aligned} u(x, 0) &= \cos(x), u(x, \frac{\pi}{2}) = 0, \\ u(0, y) &= \cos(y), u(\pi, y) = -\cos(y). \end{aligned} \quad (17)$$

Exact solution administered by,

$$u(x, y) = \cos(x) \cos(y). \quad (18)$$

$$\text{Maximum error} = \max\{| \cos(x)\cos(y) - u_{ij} | \}.$$

NUMERICAL EXPERIMENT 2 (EXPERIMENT 2)

In experiment 2, the 2<sup>nd</sup> order elliptic equation namely Poisson's equation that is considered is:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) = (x^2 + y^2)e^{xy}, \quad (19)$$

with region  $R, (x, y) \in (0, 1) \times (0, 1)$ , boundary condition given by:

$$u(x, 0) = u(0, y) = 1, u(x, 1) = e^x, u(1, y) = e^y. \quad (20)$$

Exact solution is administered by,

$$u(x, y) = e^{xy}. \quad (21)$$

$$\text{Maximum error} = \max\{| e^{xy} - u_{ij} | \}.$$

NUMERICAL EXPERIMENT 3 (EXPERIMENT 3)

In experiment 3, the 2<sup>nd</sup> order elliptic equation namely Poisson's equation is considered:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) = x, \quad (22)$$

with region  $R, (x, y) \in (0, 1) \times (0, 2)$ , boundary condition is given by:

$$\begin{aligned} u(x, 0) &= x^2, u(x, 2) = (x - 2)^2, \\ u(0, y) &= y^2, u(1, y) = (y - 1)^2. \end{aligned} \quad (23)$$

NUMERICAL EXPERIMENT 4 (EXPERIMENT 4)

In experiment 4, the 2<sup>nd</sup> order elliptic equation namely Poisson's equation that is considered is:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) = x, \quad (24)$$

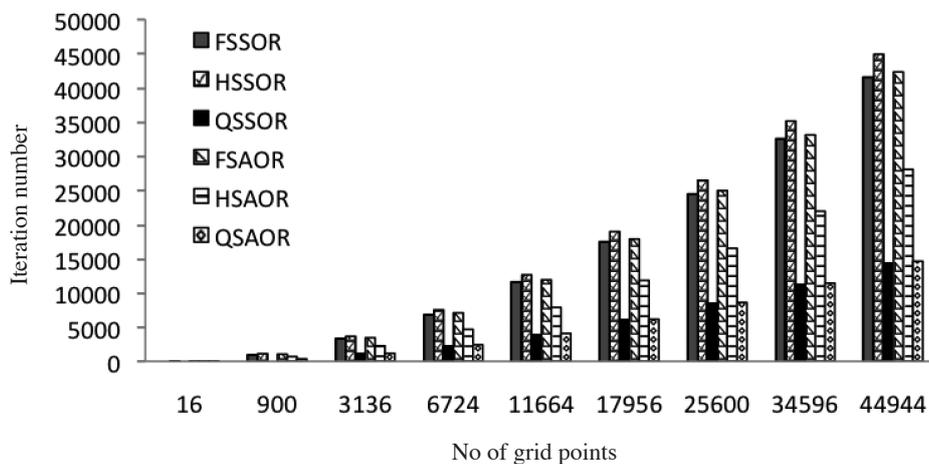


FIGURE 6. Comparison of iteration number for Experiment 2

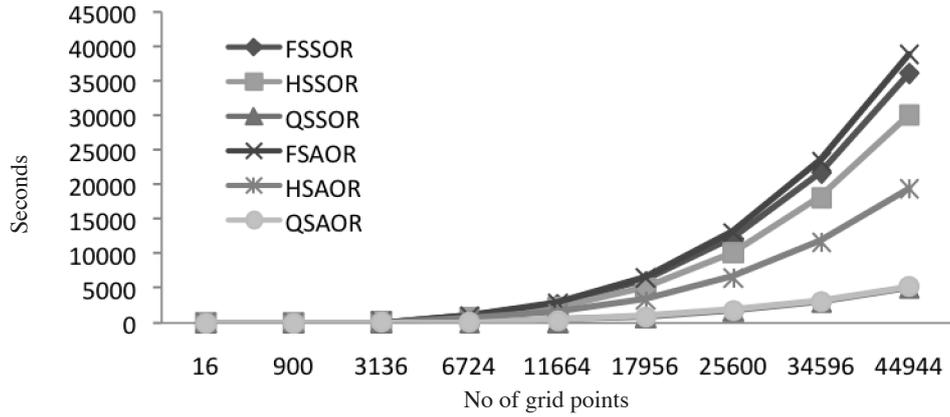


FIGURE 7. Comparison of execution time for Experiment 2

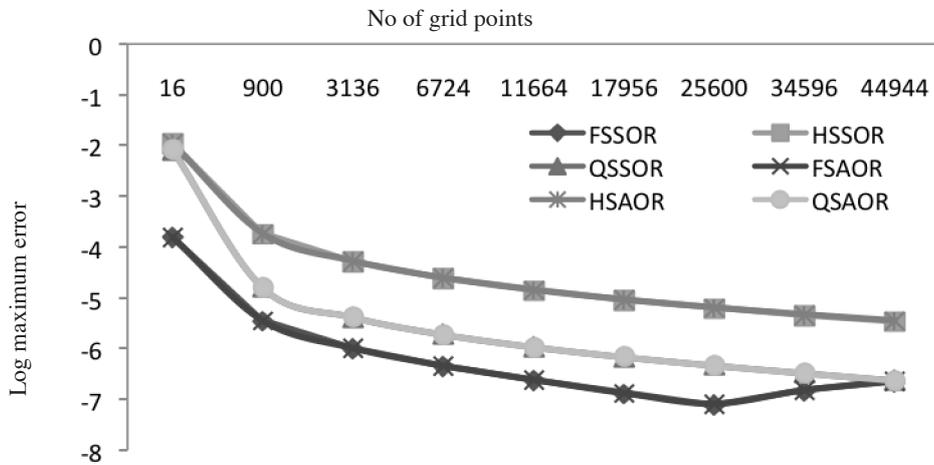


FIGURE 8. Comparison of accuracy for Experiment 2

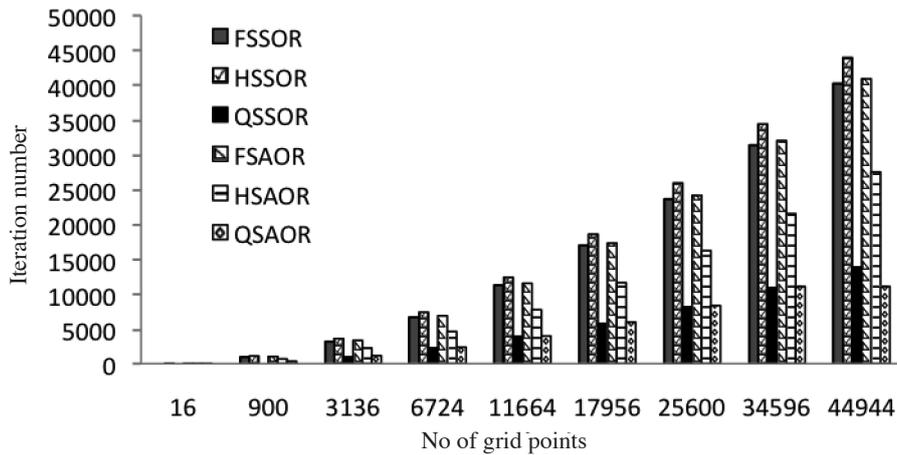


FIGURE 9. Comparison of iteration number for Experiment 3

with region  $R$ ,  $(x, y) \in (0, 1) \times (0, 1)$ , the boundary conditions are given by:

$$\begin{aligned} u(x, 0) &= \frac{1}{6}x^3, & u(x, 1) &= \frac{1}{6}x^3, \\ u(0, y) &= 0, & u(1, y) &= \frac{1}{6}. \end{aligned} \tag{25}$$

### DISCUSSION

In experiment 1, the findings demonstrate that the maximum error of full sweep is smaller than half sweep which in turn is smaller than quarter sweep case. Figure 5 shows the accuracy of FSAOR method is superior to FSSOR, HSSOR,

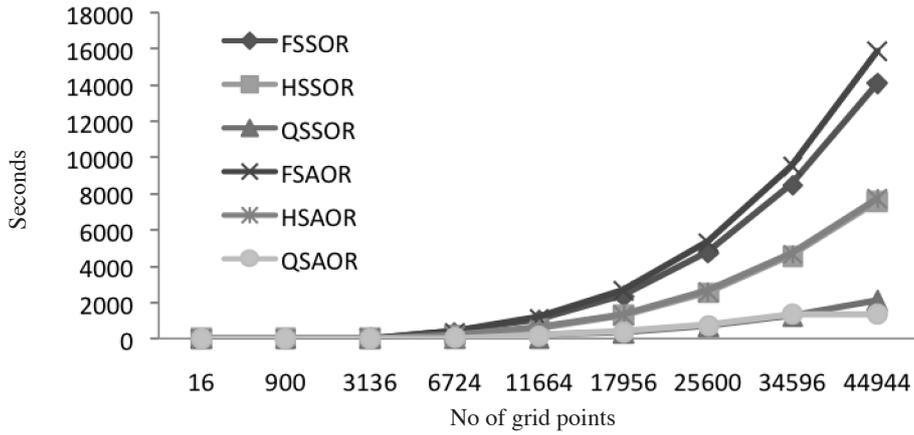


FIGURE 10. Comparison of execution time for Experiment 3

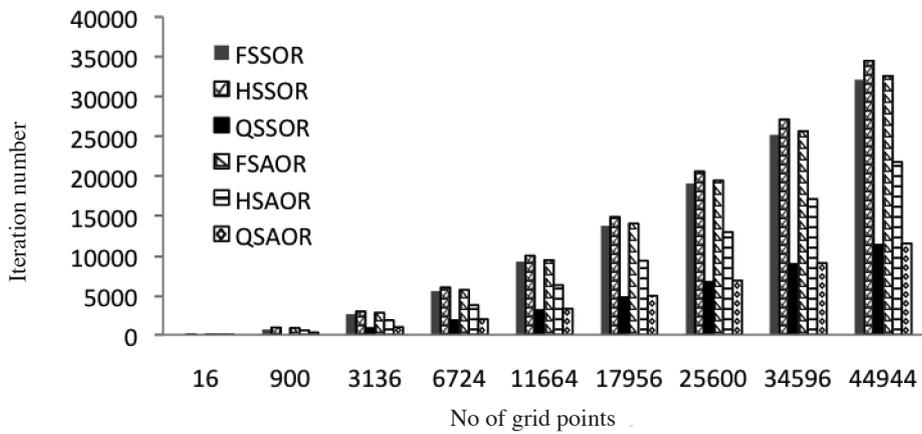


FIGURE 11. Comparison of iteration number for Experiment 4

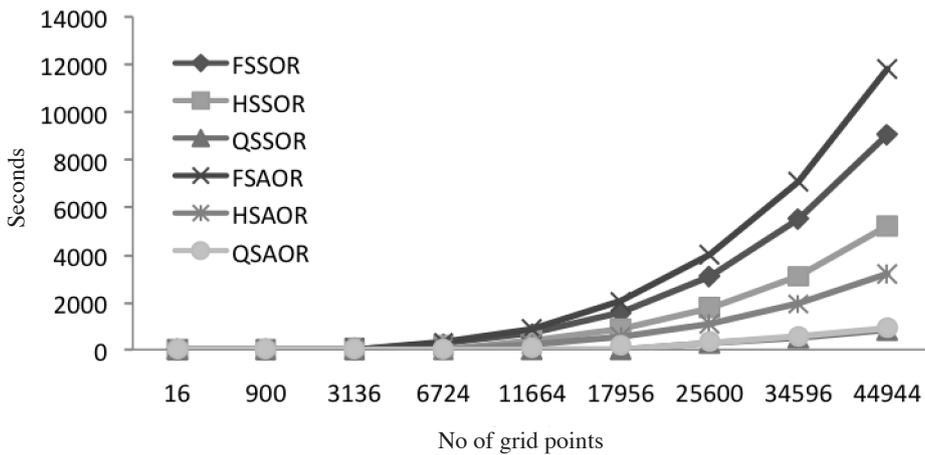


FIGURE 12. Comparison of execution time for Experiment 4

QSSOR, HSAOR and QSAOR methods. The approximate solution for HSSOR, QSSOR, HSAOR and QSAOR methods are in good agreement with FSAOR and FSSOR methods. In Figure 4, the execution time,  $t$  for HSSOR, QSSOR, HSAOR and QSAOR are faster than FSSOR method (approximately by 17.17%-100%, 85.13%-100%, 68.31%-100% and 0%-

86.56%), respectively. However, the execution time of FSAOR is slower than FSSOR (approximately 0%-3.83%) for large  $n$  and faster than FSSOR (approximately 100%) at  $n=16$  due to the slightly higher complexity of FSAOR method. Since the complexity of HSSOR is 50% less than FSSOR, HSSOR method performs faster than FSSOR

method even though the number of iterations of HSSOR is approximately more than FSSOR method (4.37%-45%) as shown in Figure 3. Furthermore, the numbers of iterations are approximately less by 62.96%-95%, 35%-67.92% and 62.62%-95% corresponding to the QSSOR, HSAOR and QSAOR as compared with FSSOR method. Besides, the number of iterations of FSAOR is approximately more by 0%-1.53% for large  $n$  and less by 5% at  $n=16$  as compared with the FSSOR method. Eventually, the HSSOR, QSSOR, HSAOR and QSAOR method are better than FSSOR method in terms of the execution time. Nevertheless, the accuracy of FSAOR is slightly better than FSSOR.

On the contrary, the findings in experiment 2 showed that the maximum error of full sweep is smaller than quarter sweep which in turn is smaller than half sweep case. The accuracy of FSAOR method is superior to FSSOR, HSSOR, QSSOR, HSAOR and QSAOR methods as reflected in Figure 8. The approximate solution for HSSOR, QSSOR, HSAOR and QSAOR methods are in good agreement with FSAOR and FSSOR methods. In Figure 7, the execution time,  $t$  for HSSOR, QSSOR, HSAOR and QSAOR faster than FSSOR method approximately by 12.99%-75%, 85.86%-100%, 46.09%-75% and 75%-86.36%, respectively. However, the execution time of FSAOR is approximately slower by 0%-9.81% for large  $n$  and faster by 75% at  $n=16$  as compared with FSSOR. Although, the number of iterations of HSSOR in Figure 6 is approximately more than FSSOR method (7.88%-36.36%), the complexity of HSSOR is only 50% from FSSOR. This explains the reason why HSSOR is faster than FSSOR method. Furthermore, the numbers of iterations are approximately less by 36.36%-65.83%, 32.55%-36.36% and 36.36%-65.44% corresponding to the QSSOR, HSAOR and QSAOR as compared with the FSSOR method. Besides, the number of iterations of the FSAOR is approximately more by 0%-1.68% for large  $n$  and less by 4.55% at  $n=16$  as compared with the FSSOR method. This is because AOR method is more complex than SOR method. Thus, the HSSOR, QSSOR, HSAOR and QSAOR method are better than FSSOR method in terms of the execution time. Nonetheless, the accuracy of FSAOR is slightly improved than FSSOR.

Based on the numerical results gathered in experiment 3, the findings showed that the execution time,  $t$  for HSSOR, QSSOR, HSAOR and QSAOR are approximately faster by 46.28%-85.71%, 85.95%-100%, 65.69%-85.71% and 85.71%-100% as compared with the FSSOR method, respectively, as shown in Figure 10. However, the execution time of FSAOR is approximately slower by 0%-0.92% for large  $n$  and faster by 85.71% at  $n=16$  as compared with FSSOR. Even though the numbers of iterations of HSSOR are approximately more than FSSOR method (7.31%-40.90%) as given in Figure 9, the complexity of HSSOR is only 50% from FSSOR. This explains the reason why HSSOR is faster than FSSOR method. Furthermore, the numbers of iterations are approximately less by 36.36%-65.81%, 31.82%-33.09% and 36.36%-65.43% corresponding to the QSSOR, HSAOR

and QSAOR as compared with the FSSOR method. Besides, the numbers of iterations of FSAOR are approximately more by 0%-1.71% for large  $n$  and less by 4.55% at  $n=16$  as compared with FSSOR method. This is because AOR method is more complex than SOR method. Thus, the HSSOR, QSSOR, HSAOR and QSAOR methods are better than FSSOR method in the aspect of execution time. However, the accuracy of FSAOR is slightly better than FSSOR.

The numerical result in experiment 4 showed that the execution time,  $t$  for HSSOR, QSSOR, HSAOR and QSAOR are faster than FSSOR approximately by 0%-46.54%, 0%-86.15%, 68.31%-100% and 0%-86.56%, respectively, as given in Figure 12. However, the execution time of FSAOR is slower than FSSOR approximately 0%-3.83% for large  $n$  and faster than FSSOR 100% at  $n=16$  due to slightly higher complexity of FSAOR method. Since the complexity of HSSOR is 50% less than FSSOR, HSSOR method performs faster than FSSOR method even though the number of iterations of HSSOR is approximately more than FSSOR method (9.23%-50%) as displayed in Figure 11. The numbers of iterations are approximately less by 35%-65.77%, 30%-31.75% and 35%-72.45% corresponding to the QSSOR, HSAOR and QSAOR as compared with FSSOR method. Besides, the numbers of iterations of FSAOR are approximately more by 0%-1.79% for large  $n$  and less by 5% at  $n=16$  as compared with FSSOR method. Thus, the HSSOR, QSSOR, HSAOR and QSAOR methods are better than FSSOR method in term of the execution time. However, the accuracy of FSAOR is slightly better than FSSOR.

## CONCLUSION

The result of experiment 1 showed that the maximum error of full sweep is the smallest, followed by half sweep and quarter sweep cases. The FSAOR method performs more accurate than FSSOR, HSSOR, QSSOR, HSAOR and QSAOR methods. Meanwhile, the result in Experiment 2 showed that the maximum error of full sweep is still the smallest. However, quarter sweep shows better performance than half sweep case. The FSAOR method performs more accurately than the others. Additionally, QSSOR and QSAOR methods perform second best among compared methods.

In term of the execution time, QSSOR and QSAOR method perform the fastest among all compared methods. HSSOR and HSAOR methods however, execute faster than their full sweep schemes. Both FSAOR and FSSOR methods need the longest execution time, since the number of iterations of QSSOR, HSAOR and QSAOR methods are less than FSSOR method. Then, the number of iterations of FSAOR and HSSOR methods are more than FSSOR method for large  $n$ . Quarter sweep is nearly 25% and half sweep is nearly 50% of complexity from full sweep case which cause the execution time performs faster.

In conclusion, by analysing both accuracy and execution time, quarter sweep schemes (QSSOR and QSAOR) give the best performance compared with others.

## ACKNOWLEDGEMENT

We acknowledge the grant from Young Researchers Grant UKM-GGPM-ICT-110-2010. We also thank the referees for the suggestions that improved the paper.

## REFERENCES

- Ali, N.H. & Evans, D.J. 2004. Preconditioned rotated iterative methods in the solution of elliptic partial differential equation. *International Journal of Computer Mathematics* 81(9): 1163-1174.
- Ali, N.H. & Sam, T.L. 2006. On implementing several krylov preconditioned iterative schemes on rotated grid. *2nd Annual Proceedings of the IMT-GT Regional Conference on Mathematics, Statistics and Application*.
- Hadjidimos, A. 1978. Accelerated overrelaxation method. *Mathematics of Computation* 32: 149-157.
- Lee, S.C. 2007. Preconditioned 5 rotated scheme in solving the Poisson's equation. *17th Annual Proceedings of the Simposium Kebangsaan Sains Matematik* pp.154-159.
- Moussavi, S. 2009. One step closer to an optimal two-parameter SOR method - A geometric approach. *Journal of Mathematical Sciences* 21(1): 56-64.
- Othman, M., Rakhimov, S.I. & Sulaiman, J. 2009. An AOR four points modified explicit group iterative algorithms for solving 2D Poisson equation. *5th Annual Proceedings of the Asian Mathematical Conference* pp. 276-279.
- Sulaiman, J., Othman, M. & Hasan, M.K. 2006. An optimal ordering strategy of the point iterative algorithm for solving 2D convection-diffusion problem. *Borneo Science* 18: 1-8.
- Sulaiman, J., Othman, M. & Hasan, M.K. 2008a. Half-sweep algebraic multigrid (HSAMG) method applied to diffusion equations in modeling. *Simulation and Optimization of Complex Processes*, Springer-Verlag, Berlin Heidelberg. pp. 547-556.
- Sulaiman, J., Othman, M. & Hasan, M.K. 2007. Red-black EDGSOR iterative method using triangle finite element approximation for 2D poisson equations. *LNCS 4707 Part III*: 298-308.
- Sulaiman, J., Saudi, A., Abdullah, M.H., Hasan, M.K. & Othman, M. 2008b. Quarter-sweep arithmetic mean algorithm for water quality model, *Proceeding International Symposium on Information Technology* 3: 1859-1863.
- Wang, G., Wen, H., Li, L. & Li, X. 2011. Convergence of GAOR method for doubly diagonally dominant matrices. *Journal of Applied Mathematics and Computation* 217: 7509-7514.
- Young, D.M. 1950. Iterative methods for solving partial difference equations of elliptic type, A Ph.D. Thesis of Historical Importance, Harvard University Cambridge. Mass (unpublished).

School of Information Technology  
Faculty of Information Science and Technology  
Universiti Kebangsaan Malaysia  
43650 Bangi, Selangor D.E.  
Malaysia

\*Corresponding author; email: fido\_ng87@yahoo.com

Received: 19 July 2012

Accepted: 14 November 2012