

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Recent Development of Fast Numerical Solver for Elliptic Problem

Mohammad Khatim Hasan¹, Jumat Sulaiman²,
Samsul Ariffin Abdul Karim³, and Mohamed Othman⁴

¹*School of Information Technology, Universiti Kebangsaan Malaysia*

²*School of Science and Technology, Universiti Malaysia Sabah*

³*Department of Fundamental and Applied Science, Universiti Teknologi Petronas*

⁴*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
Malaysia*

1. Introduction

Most elliptic solvers developed by researchers need long processing time to be solved. This is due to the complexity of the methods. The objective of this paper is to present new finite difference and finite element methods to overcome the problem.

Solving scientific problems mathematically always involved partial differential equations. Two recommended common numerical methods are mesh-free solutions (Belytschko et al, 1996; Zhu 1999; Yagawa & Furukawa, 2000) and mesh-based solutions. The mesh-based solutions can be further classified as finite difference method, finite element method, boundary element method, and finite volume method. These methods have been widely used to construct approximation equations for scientific problems.

The developments of numerical algorithms have been actively done by researchers. Evans and Biggins (1982) have proposed an iterative four points Explicit Group (EG) for solving elliptic problem. This method employed blocking strategy to the coefficient matrix of the linear system of equations. By implementing this strategy, four approximate equations are evaluated simultaneously. This scenario speed up the computation time of solving the problem compared to using point based algorithms.

At the same time, Evans and Abdullah (1982) utilized the same concepts to solve parabolic problem. Four years later, the concept has been further extended to develop two, nine, sixteen and twenty five points EG (Yousif & Evans, 1986a). These EG schemes have been compared to one and two lines methods. As the results of comparison, the EG solve the problem efficiently compared to the lines methods.

Utilizing higher order finite difference approximation, a method called Higher Order Difference Group Explicit (HODGE) was developed (Yousif & Evans, 1986b). This method have higher accuracy than the EG method. Abdullah (1991) modified the EG method by using rotated approximation scheme. The rotated scheme is actually rotate the ordinary computational molecule by 45° to the left. By rearranging the new computational molecule on the solution domain, only half of the total nodes are solved iteratively. The other half can be solved directly using the ordinary computational molecule. This method was named

Explicit Decoupled Group (EDG). He use this new method to solve the two dimensional Poisson problem and was proven to be faster solver than the EG method by 50%. The performance of the EDG method was further tested by Ibrahim. He implements the method to solve boundary value problem (Ibrahim, 1993) and two dimensional diffusion problem (Ibrahim & Abdullah, 1995). The EDG method was then extended to six and nine points (Yousif & Evans, 1995).

This fast Poisson solver have been challenged by a method called four points Modified Explicit Group (MEG) method (Othman & Abdullah, 2000). The concept utilised in MEG method was created from modification of concept used in EDG method. In the MEG method, only a quarter of node points are solved iteratively, and the remaining points are solved directly using the standard and rotated algorithm (Othman & Abdullah, 2000). MEG method has successfully saving about 50% of EDG computational time and 75% of EG computational time. An additional advantage is the MEG method also has higher accuracy compared to EDG method.

In this chapter, we will demonstrate newly develop finite difference and finite element method based on the concept mentioned above for the solution of elliptic problem.

2. Finite difference method with red black ordering

We developed two practical finite difference techniques utilising the concept proposed by Abdullah (1991) and Othman & Abdullah (2000) for solving elliptic problem.

Consider the 2D Hemholtz equation as follows.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \wp U = f(x, y), (x, y) \in [a, b] \times [a, b], \quad (1)$$

Subject to Dirichlet boundary conditions

$$U(x, a) = g_1(x), a \leq x \leq b,$$

$$U(x, b) = g_2(x), a \leq x \leq b,$$

$$U(a, y) = g_3(y), a \leq x \leq b,$$

$$U(b, y) = g_4(y), a \leq x \leq b.$$

In this article, we will only consider uniform nodes. Utilising the concept from MEG (Othman & Abdullah, 2000), we develop a method called Quarter Sweep Successive Over-Relaxation using red black ordering strategy(QSSOR-RB). Utilising the concept in EDG (Abdullah, 1991) and the red black ordering strategy, we develop a method called Half Sweep Successive Over Relaxation (HSSOR-RB). Employing the Successive Over-Relaxation (SOR) method using the same concept in MEG, QSSOR-RB method only solve quarter node point iteratively and utilising the concept in EDG, HSSOR method only solve half of the node points iteratively by SOR method. Beside that the nodes are arranged in a Red-Black ordering manner (Figure 3).

There are so many approached can be used to approximate problem (1). For instance, Rosser (1975) and Gupta et al. (1997) have proposed low and high order schemes. Both schemes can be rewritten in the forms of systems of linear equations. However, both system of equations will have distinct properties of coefficient matrix from each other.

Based on second order schemes, the full and quarter sweeps approximation equations can be generally stated as

$$U_{i-p,j} + U_{i+p,j} + U_{i,j-p} + U_{i,j+p} - (4 + \wp(ph)^2)U_{i,j} = (ph)^2 f_{i,j}. \quad (2)$$

Value of $p=1$ represent the full sweep scheme and is used to solve all black nodes in discrete solution domain given in Figure 1a iteratively. While $p=2$ represent the quarter sweep schemes and is used to solve all black nodes in discrete solution domain given in Figure 1b iteratively.

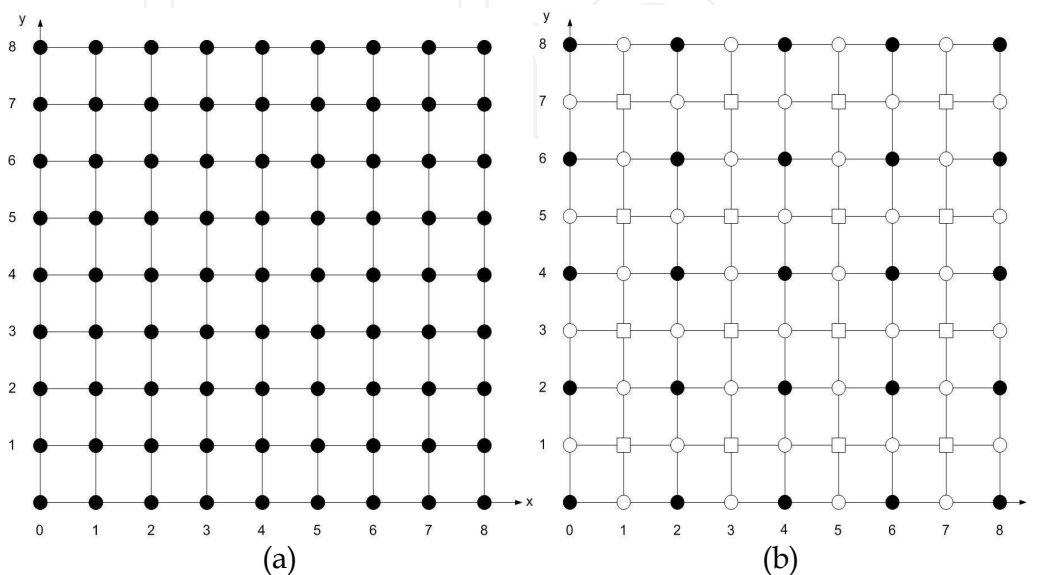


Fig. 1. Discrete solution domain for (a) full sweep and (b) quarter sweep schemes

By employing the concept used in EDG (Abdullah, 1991), the five points rotated finite difference approximation equation can be formed. The transformation processes are as follows.

$$\begin{aligned} i, j \pm 1 &\rightarrow i \pm 1, j \pm 1, \\ i \pm 1, j &\rightarrow i \pm 1, j \mp 1, \\ \Delta x, \Delta y &\rightarrow \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{2}h, \Delta x = \Delta y = h \end{aligned}$$

Applying this transformation, approximation equation (2) can be rewritten as

$$U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - (4 + 2\wp h^2)U_{i,j} = 2h^2 f_{i,j}. \quad (3)$$

The approximation equation (3) is applied on solution domain displayed in Figure 2 to solve all black nodes iteratively. The white box nodes in Figure 1b are solved via

$$U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - (4 + 2\wp h^2)U_{i,j} = 2h^2 f_{i,j}.$$

directly and the white bullet nodes in Figures 1b and 2 by

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - (4 + \wp(ph)^2)U_{i,j} = (ph)^2 f_{i,j},$$

directly.

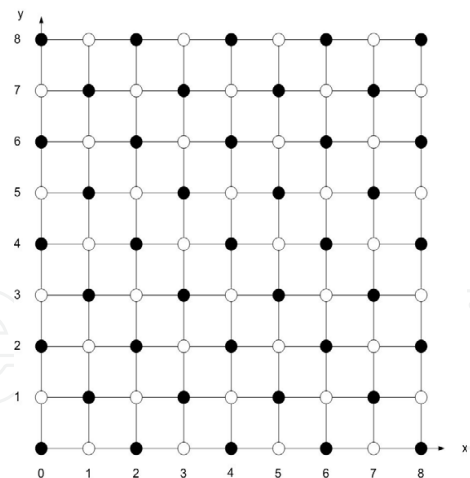


Fig. 2. Discrete solution domain for half sweep scheme

Red-Black ordering strategies have been shown to accelerate the convergence of many numerical algorithm (Parter, 1998; Evans & Yousif, 1990; Zhang, 1996). Hence, we apply this ordering strategy to further increase the speed of our computation. The Implementation of the Red-Black ordering strategy are shown in Figure 3.

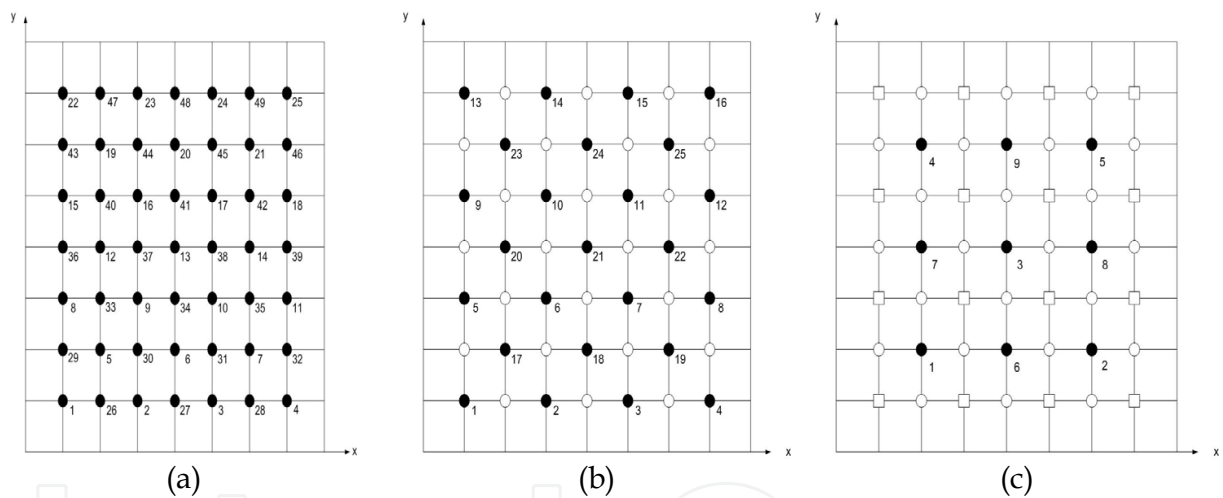


Fig. 3. Implementation of the RB ordering strategy for (a) full-sweep, (b) half-sweep, and (c) quarter-sweep cases.

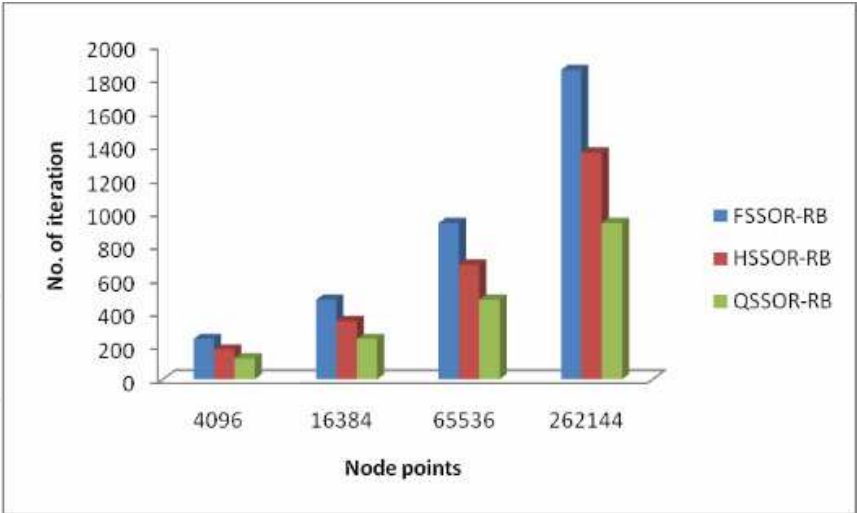
The efficiency of the FSSOR-RB, HSSOR-RB and QSSOR-RB are analysed via the following two dimensional Helmholtz equation.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \rho U = 6 - \alpha(2x^2 + y^2), (x,y) \in [0,1] \times [0,1],$$

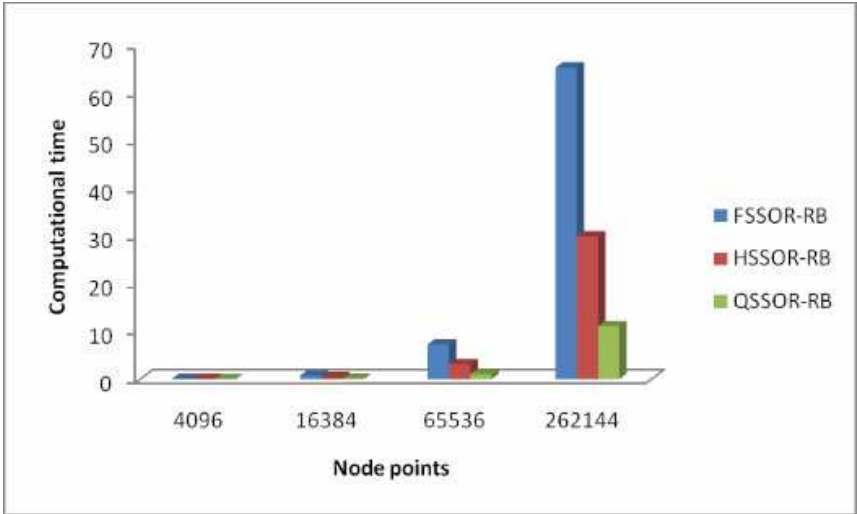
with the boundaries and exact solution are defined by

$$U(x,y) = 2x^2 + y^2, 0 \leq x,y \leq 1.$$

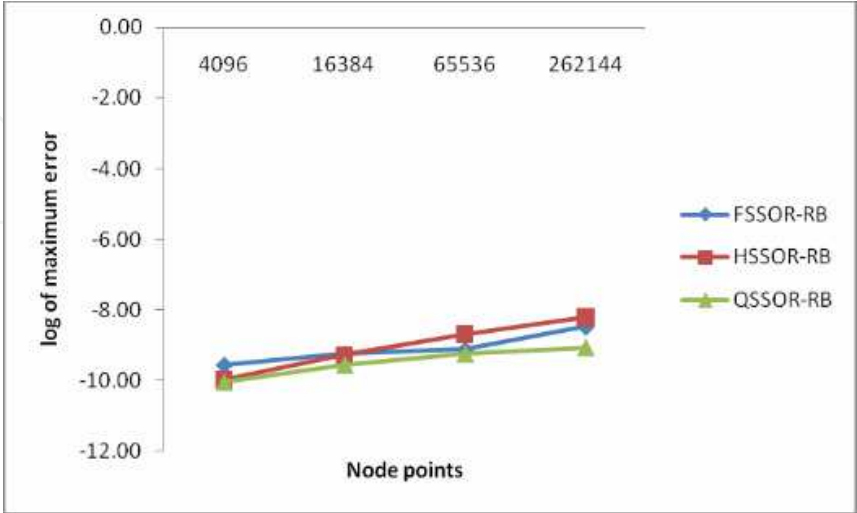
The convergence criteria considered in these experiments is $\varepsilon = 10^{-10}$. All results of numerical experiments are displayed in Figures 4 to 8.



(a)

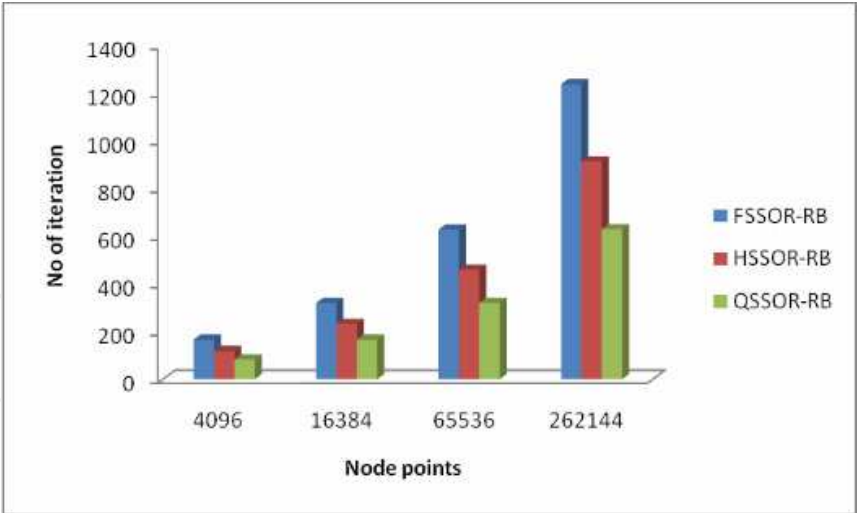


(b)

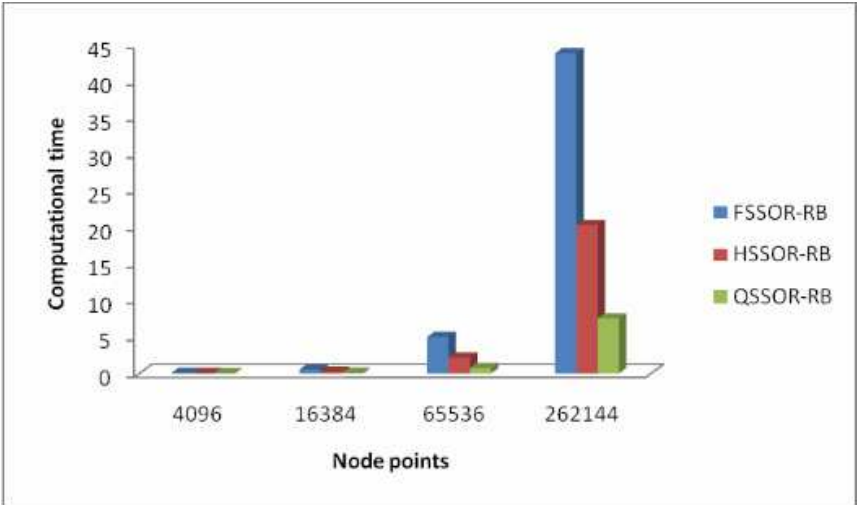


(c)

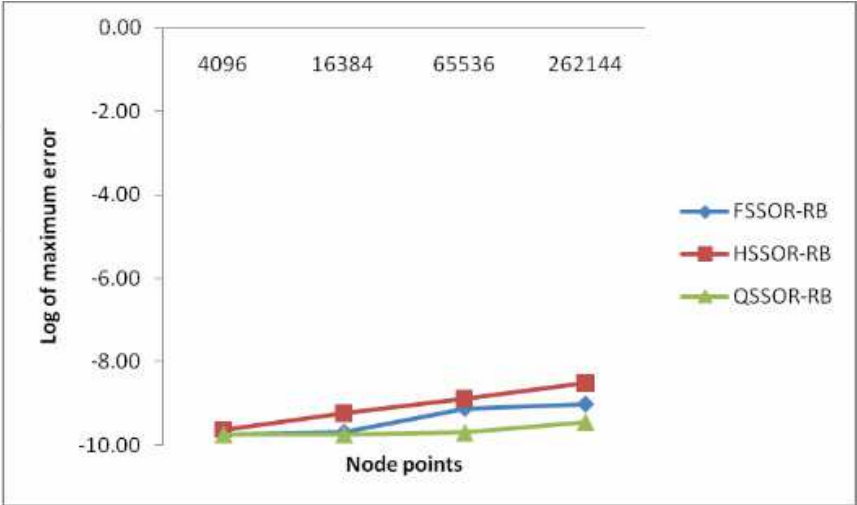
Fig. 4. Comparison of (a) iteration number, (b) computational time, and (c) accuracy for FSSOR-RB, HSSOR-RB and QSSOR-RB ($\alpha=0$)



(a)

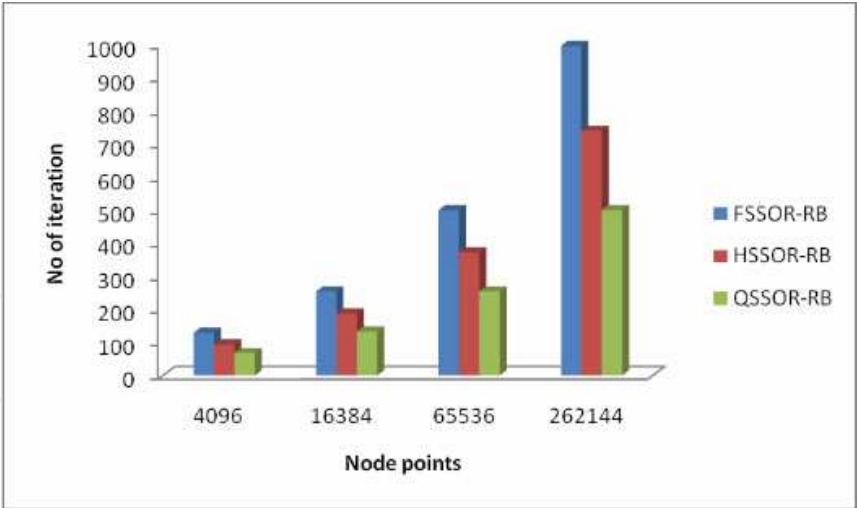


(b)

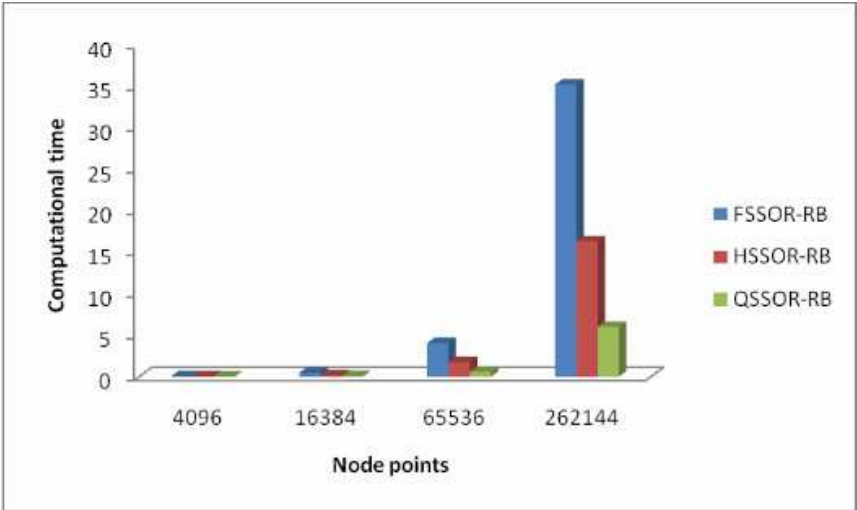


(c)

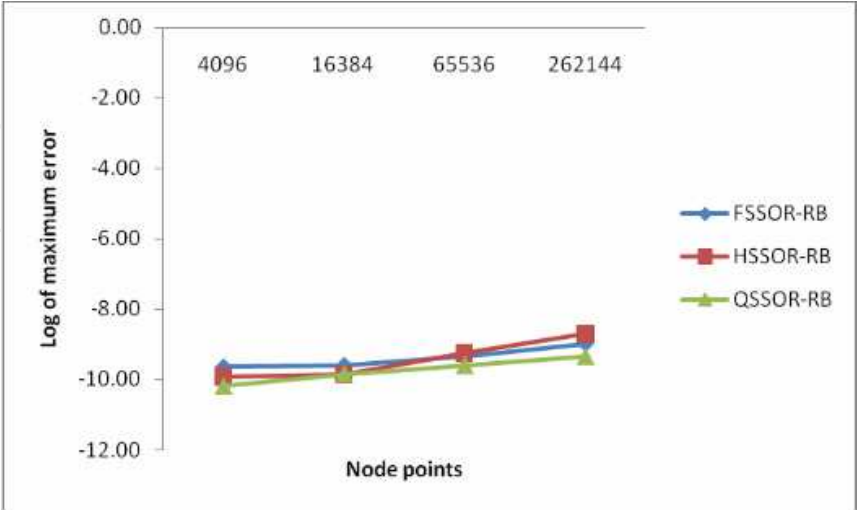
Fig. 5. Comparison of (a) iteration number, (b) computational time, and (c) accuracy for FSSOR-RB, HSSOR-RB and QSSOR-RB ($\alpha=25$)



(a)

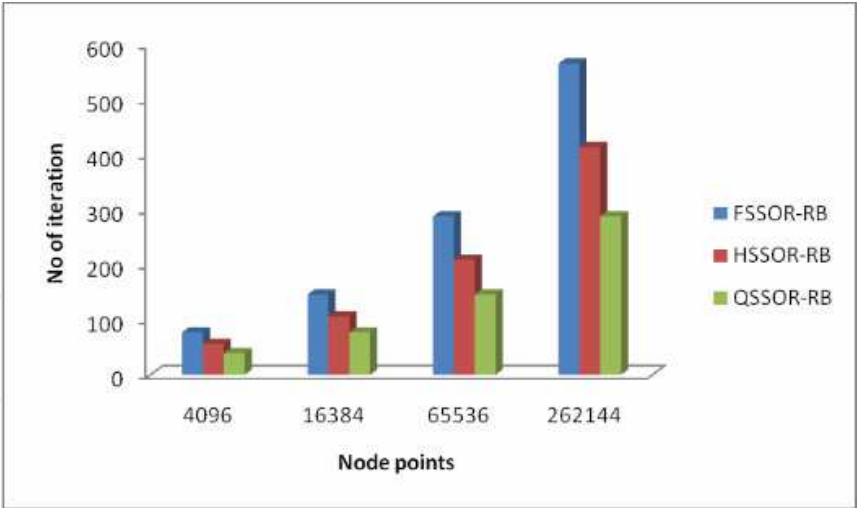


(b)

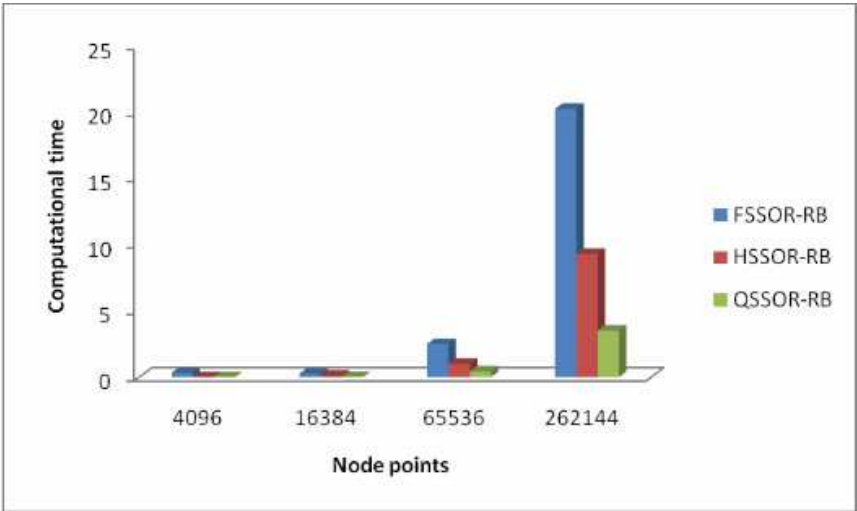


(c)

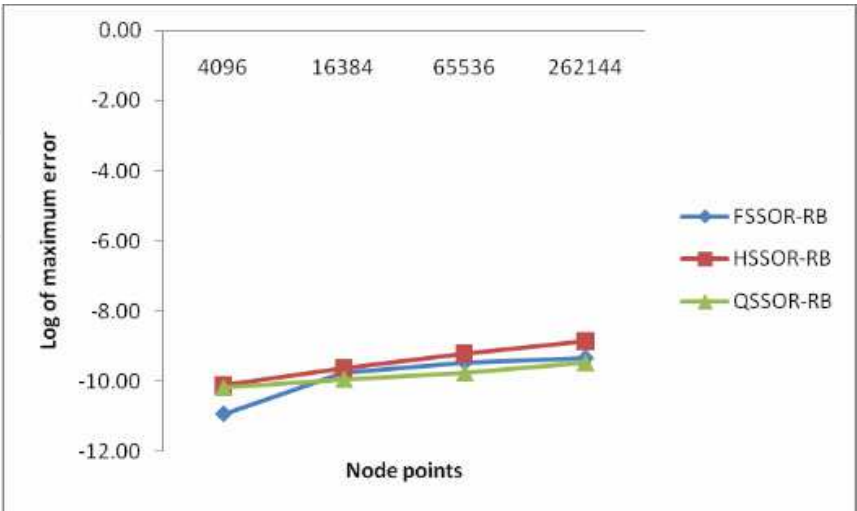
Fig. 6. Comparison of (a) iteration number, (b) computational time, and (c) accuracy for FSSOR-RB, HSSOR-RB and QSSOR-RB ($\alpha=50$)



(a)

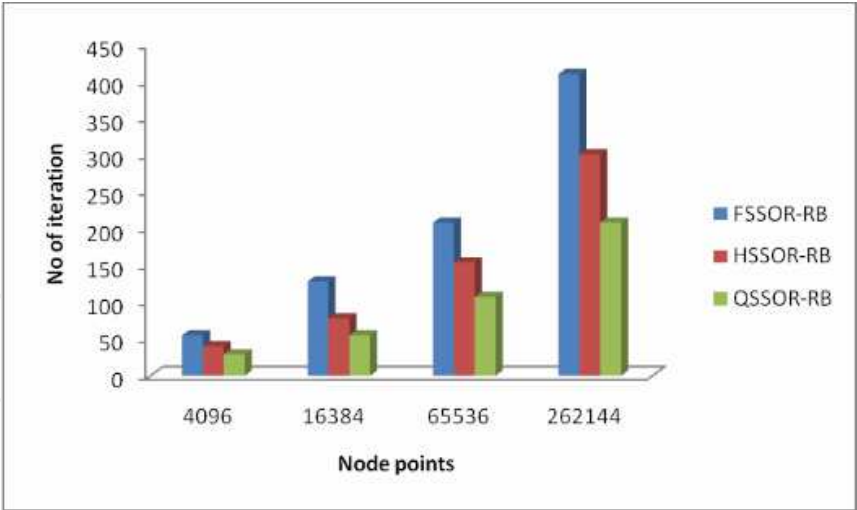


(b)

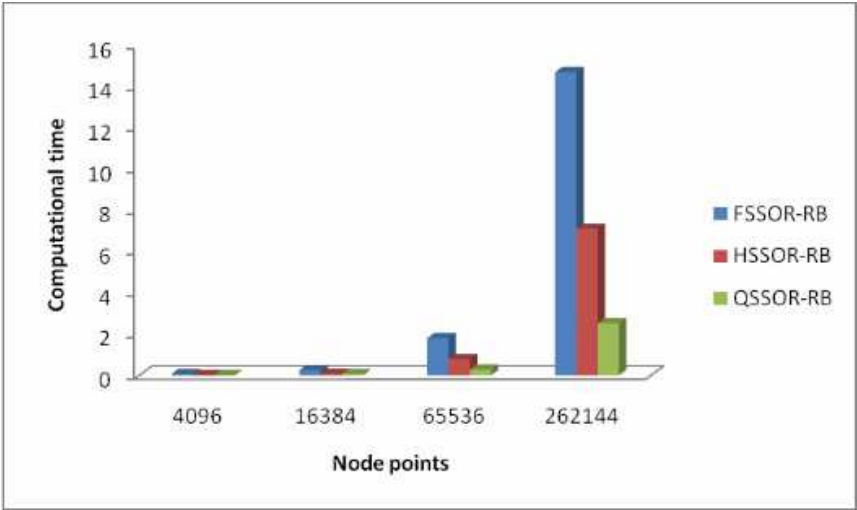


(c)

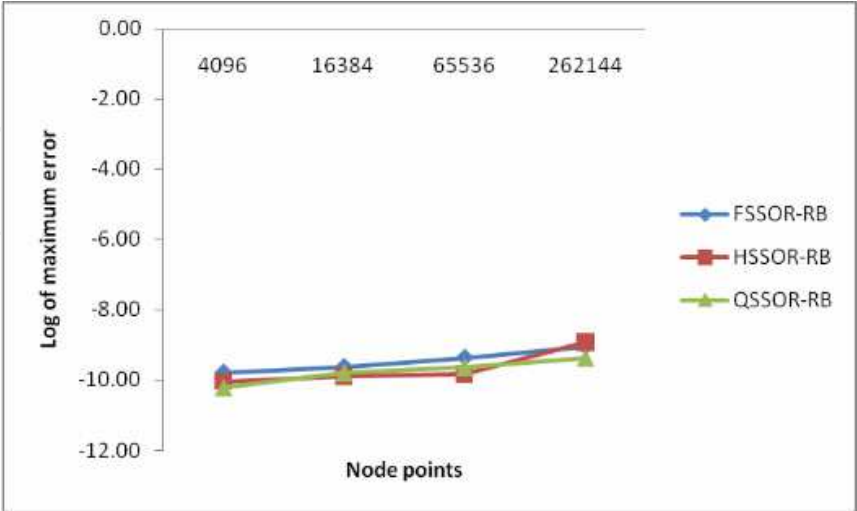
Fig. 7. Comparison of (a) iteration number, (b) computational time, and (c) accuracy for FSSOR-RB, HSSOR-RB and QSSOR-RB ($\alpha=200$)



(a)



(b)



(c)

Fig. 8. Comparison of (a) iteration number, (b) computational time, and (c) accuracy for FSSOR-RB, HSSOR-RB and QSSOR-RB ($\alpha=400$)

Figures 4 to 8 have shown that the results of experiments for FSSOR-RB, HSSOR-RB and QSSOR-RB methods. The results not only compares between these three methods, but also the impact of solving more node points on the iteration, computational time and accuracy of all three methods.

Results display in Figures 4(a) to 8(a) show that the numbers of iterations are impacted by the number of node points solved. The number of iterations and node points solved are the measure of complexity of methods since it refers to the number of function evaluation for each methods and problems. The number of node points solved in all experiments are 4096, 16384, 65536 and 262144. The ratio is 1: 4: 16: 64. However the ratio of number of iteration is about 1: 2: 4: 8. It means that increasing the problem sized considered by $4^i, i = 0, 1, 2, 3$ increase the iterations by $2^i, i = 0, 1, 2, 3$. It is the power of two relationships. By making the QSSOR-RB as the basis of comparison, the FSSOR-RB needs around 1.89 to 2.33 times and HSSOR-RB needs around 1.37 to 1.48 times number of iterations compared to QSSOR-RB. This means that FSSOR-RB is two times more complex than QSSOR-RB and HSSOR-RB is 1.5 times more complex than QSSOR-RB. This is equivalent to what we are expected since QSSOR-RB only solve a quarter of the node points in solution domain iteratively and HSSOR-RB only solve half of the node points in the solution iteratively, while FSSOR-RB have to solve every node in the solution domain iteratively.

Results displayed in Figures 4(b) to 8(b) support our description in the above paragraph. From the theoretical form of view, higher complexity method needs more computational time to solve problem. These scenarios are shown in Figures 4(b) to 8(b). However the effect of the complexity is different following the α value used. Figure 4(b) shows that FSSOR-RB needs 5 to 6.86 times more computational time compared to QSSOR-RB while HSSOR-RB only needs 2.69 to 3.5 times for $\alpha = 0$. However, for $\alpha=25$, FSSOR-RB needs 3 to 6.9 times more computational time to solve the problem compared to QSSOR-RB, while HSSOR-RB only need 2 to 3 times more computational time compared to QSSOR-RB (Refer Figure 5(b)). For $\alpha=50$, the FSSOR-RB needs 2.5 to 7.23 times more computational time compared to QSSOR-RB and HSSOR-RB only need 1.5 to 3.09 times more computational time. As for bigger α value, the interval of computational time ratio of HSSOR-RB as compared to QSSOR-RB is narrowing and for FSSOR-RB becoming wider.

Figures 4(c) to 8(c) compares the accuracy of FSSOR-RB, HSSOR-RB and QSSOR-RB. These figures show that the accuracy of all methods are almost similar except for $\alpha=25$ (refer Figure 5(c)). The figure shows that the QSSOR-RB has the highest accuracy, followed by FSSOR-RB and the last one is the HSSOR-RB.

3. Finite element method with red black ordering

In this subtopic, we will explain the development of our finite element class of method based on Galerkin scheme using triangle element discretization. Other finite element schemes are subdomain, collocation, least-square and moment.

Consider the two dimensional Poisson equation as follows.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y), (x, y) \in [a, b] \times [a, b]. \quad (4)$$

The Dirichlet boundary conditions are given by

$$\begin{aligned}
 U(x, a) &= g_1(x), a \leq x \leq b, \\
 U(x, b) &= g_2(x), a \leq x \leq b, \\
 U(a, y) &= g_3(y), a \leq x \leq b, \\
 U(b, y) &= g_4(y), a \leq x \leq b.
 \end{aligned}$$

A networks of triangle elements need to be build in order to derived the triangle element approximation equations for problem (4). The general approximation of the function, $U(x, y)$ in the form of interpolation function is given by

$$\underline{U}^{[e]}(x, y) = N_1(x, y)U_1 + N_2(x, y)U_2 + N_3(x, y)U_3. \quad (5)$$

The shape function can be stated as

$$N_k(x, y) = \frac{1}{|A|} (a_k + b_k x + c_k y), k = 1, 2, 3 \quad (6)$$

where,

$$\begin{aligned}
 |A| &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2), \\
 a_1 &= x_2y_3 - x_3y_2, a_2 = x_3y_1 - x_1y_3, a_3 = x_1y_2 - x_2y_1, \\
 b_1 &= y_2 - y_3, b_2 = y_3 - y_1, b_3 = y_1 - y_2, \\
 c_1 &= x_3 - x_2, c_2 = x_1 - x_2, c_3 = x_2 - x_1.
 \end{aligned}$$

The first order partial derivatives for the shape functions are given as follows.

$$\frac{\partial}{\partial x} (N_k(x, y)) = \frac{b_k}{|A|}, k = 1, 2, 3 \quad (7)$$

in x direction and

$$\frac{\partial}{\partial y} (N_k(x, y)) = \frac{c_k}{|A|}, k = 1, 2, 3 \quad (8)$$

in y direction.

Based on the definition of hat function, the approximation functions for problem (4) are given as follows.

$$\underline{U}(x, y) = \sum_{r=0}^m \sum_{s=0}^m R_{r,s}(x, y)U_{r,s} \quad (9)$$

for full sweep case,

$$\underline{U}(x, y) = \sum_{r=0(2)}^m \sum_{s=0(2)}^m R_{r,s}(x, y)U_{r,s} + \sum_{r=1(2)}^{m-1} \sum_{s=1(2)}^{m-1} R_{r,s}(x, y)U_{r,s} \quad (10)$$

for half sweep case, and

$$\underline{U}(x, y) = \sum_{r=0(2)}^m \sum_{s=0(2)}^m R_{r,s}(x, y) U_{r,s} \quad (11)$$

for quarter sweep case.

Next, let consider the Galerkin residual method as follows.

$$\iint_D R_{i,j}(x, y) E(x, y) dx dy = 0, i, j = 0, 1, 2, \dots, m \quad (12)$$

With, $E(x, y) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - f(x, y)$ is the residual function. Applying the Green theorem to equation (9) yields

$$\oint_{\lambda} \left(-R_{i,j}(x, y) \frac{\partial U}{\partial y} dx + R_{i,j}(x, y) \frac{\partial U}{\partial x} dy \right) - \int_a^b \int_a^b \left(\frac{\partial R_{i,j}}{\partial x} \frac{\partial U}{\partial x} + \frac{\partial R_{i,j}}{\partial y} \frac{\partial U}{\partial y} \right) dx dy = F_{i,j} \quad (13)$$

with $F_{i,j} = \int_a^b \int_a^b R_{i,j}(x, y) f(x, y) dx dy$.

By replacing equation (7), (8) and the boundary conditions into problem (4), can be shown that equation (13) will generate a linear system for any cases. The linear system can be stated as

$$-\sum \sum K_{i,j,r,s}^* U_{r,s} = \sum \sum C_{i,j,r,s}^* f_{r,s} \quad (14)$$

with

$$K_{i,j,r,s}^* = \int_a^b \int_a^b \left(\frac{\partial R_{i,j}}{\partial x} \frac{\partial R_{r,s}}{\partial x} \right) dx dy + \int_a^b \int_a^b \left(\frac{\partial R_{i,j}}{\partial y} \frac{\partial R_{r,s}}{\partial y} \right) dx dy$$

and

$$C_{i,j,r,s}^* = \int_a^b \int_a^b (R_{i,j}(x, y) R_{r,s}(x, y)) dx dy.$$

In stencil form, the full, half, and quarter sweep can be stated as follows.

- Full sweep stencil

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} U_{i,j} = \frac{h^2}{12} \begin{bmatrix} & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & \end{bmatrix} f_{i,j}.$$

This stencil is applied on solution domain with triangle element displayed in Figure 9.

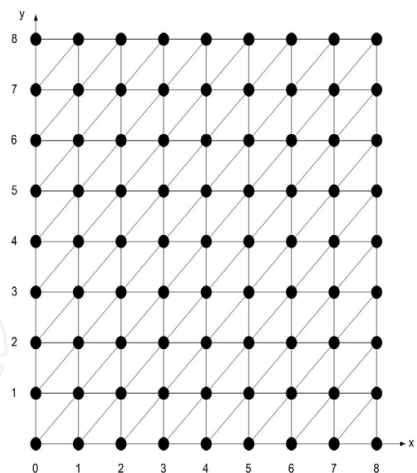


Fig. 9. Solution domain with triangle element discretization for full sweep scheme

- Half sweep stencil

$$\begin{bmatrix} 1 & 1 & \\ & -4 & 0 \\ 1 & 1 & \end{bmatrix} U_{i,j} = \frac{h^2}{6} \begin{bmatrix} 1 & 1 & \\ & 5 & 1 \\ 1 & 1 & \end{bmatrix} f_{i,j}, i = 1$$
$$\begin{bmatrix} 1 & 1 & \\ & -4 & 0 \\ 1 & 1 & \end{bmatrix} U_{i,j} = \frac{h^2}{6} \begin{bmatrix} 1 & 1 & 1 \\ & 6 & 1 \\ 1 & 1 & 1 \end{bmatrix} f_{i,j}, i \neq 1, n$$
$$\begin{bmatrix} 1 & 1 & \\ & -4 & 1 \\ 1 & 1 & \end{bmatrix} U_{i,j} = \frac{h^2}{6} \begin{bmatrix} 1 & 1 & 1 \\ & 5 & \\ 1 & 1 & 1 \end{bmatrix} f_{i,j}, i = n$$

This stencil is applied on solution domain with triangle element displayed in Figure 10.

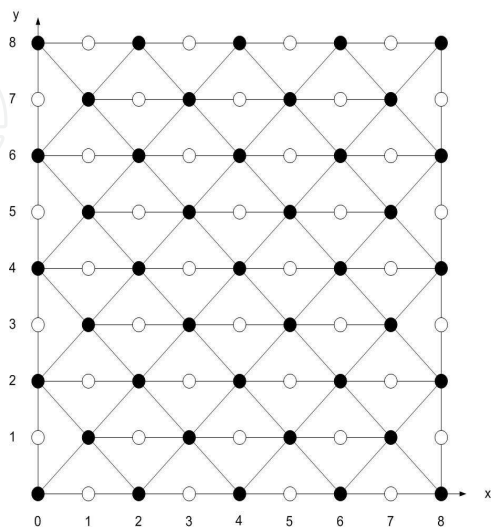


Fig. 10. Solution domain with triangle element discretization for half sweep scheme

- Quarter Sweep stencil

$$\begin{bmatrix} & & 1 & & \\ & & 0 & & \\ 1 & 0 & 4 & 0 & 1 \\ & & 0 & & \\ & & 1 & & \end{bmatrix} U_{i,j} = \frac{h^2}{3} \begin{bmatrix} 0 & 1 & 1 \\ & 0 & \\ 1 & 0 & 6 & 0 & 1 \\ & 0 & \\ 1 & 1 & 0 \end{bmatrix} f_{i,j}$$

This stencil is applied on solution domain with triangle element displayed in Figure 11.

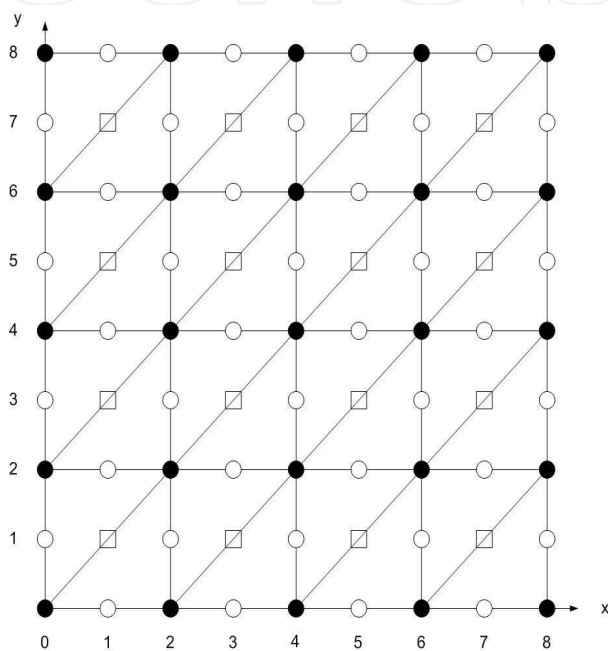


Fig. 11. Solution domain with triangle element discretization for quarter sweep scheme

All full sweep, half sweep and quarter sweep methods utilised the same red black ordering as the previous finite difference method in section 2 applied (refer Figure 3). The performance of the full sweep, half sweep and quarter sweep Gauss Seidel schemes using triangle element discretization based on Galerkin scheme are analysed for the following two dimensional Poisson equation (Abdullah, 1991).

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = (x^2 + y^2) e^{xy}, \quad (x,y) \in [0,1] \times [0,1].$$

The boundaries and the exact solution are given as follows.

$$U(x,y) = e^{xy}, \quad 0 \leq x,y \leq 1.$$

The convergence criteria considered in these experiments is $\varepsilon = 10^{-10}$. All results of numerical experiments are displayed in Figure 12, 13 and 14.

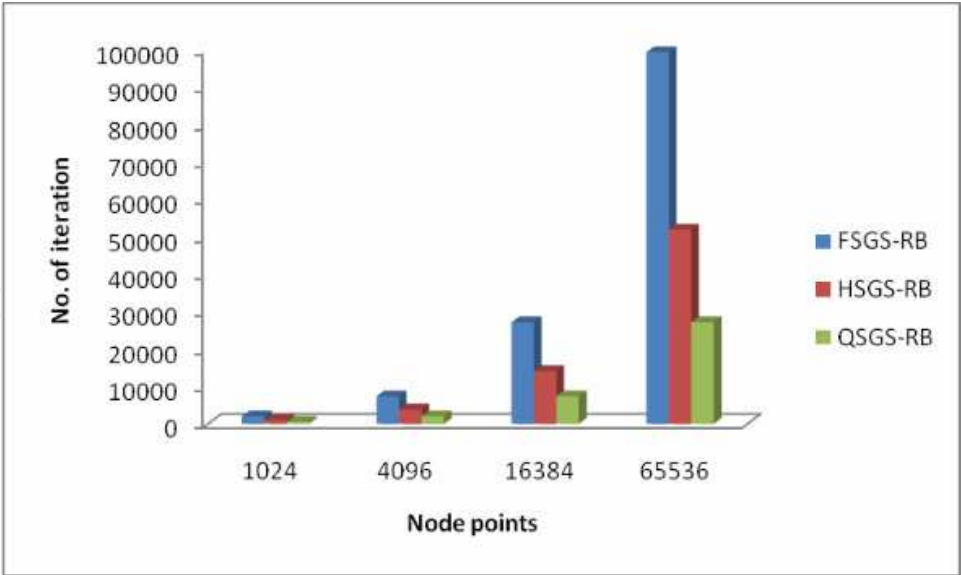


Fig. 12. Number of iterations for all compared methods.

Figures 12 to 14 have shown that the full sweep, half sweep, and quarter sweep triangle element approximation equations based on the Galerkin scheme are fast and accurate algorithms. The findings in Figure 12 shows that numbers of iteration needed by FSGS-RB are almost four times compared to QSGS-RB, while HSGS-RB is almost two times compared to QSGS-RB. The impact of increasing the number of node points to number of iterations seems too significant. In this experiment the ratio of number of points studied is 1: 4: 16: 64. However, the increment of numbers of point solves also increase the number of iterations with the ratio of 1: 4: 14: 50. This means that increasing the number of points does increase the complexity or increase the number of function evaluation. This behavior is as expected since its follows the theoretical explanation.

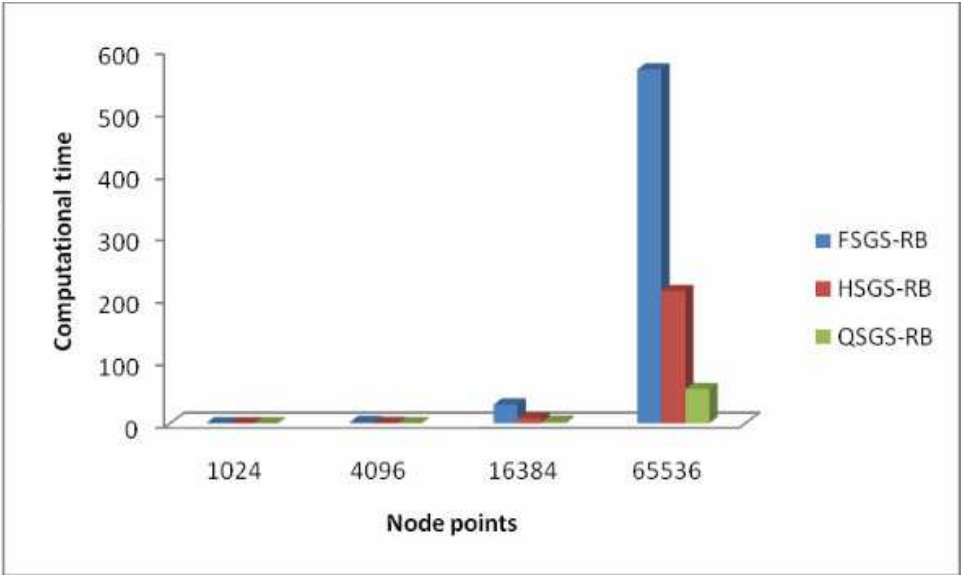


Fig. 13. Computational time in seconds for all compared methods.

Figure 13 clearly shows that QSGS-RB compute faster than the other two methods (HSGS-RB and FSGS-RB). The FSGS-RB needs 6.5 to 13.2 times more computational time compared

to QSGS-RB, while HSGS-RB only need 1.5 to 3.9 times more computational time compared to QSGS-RB. This is because QSGS-RB only solved a quarter of node points iteratively, while HSGS-RB only solved half of the node points iteratively. However, the accuracy of both QSGS-RB and HSGS-RB are lower than FSGS-RB.

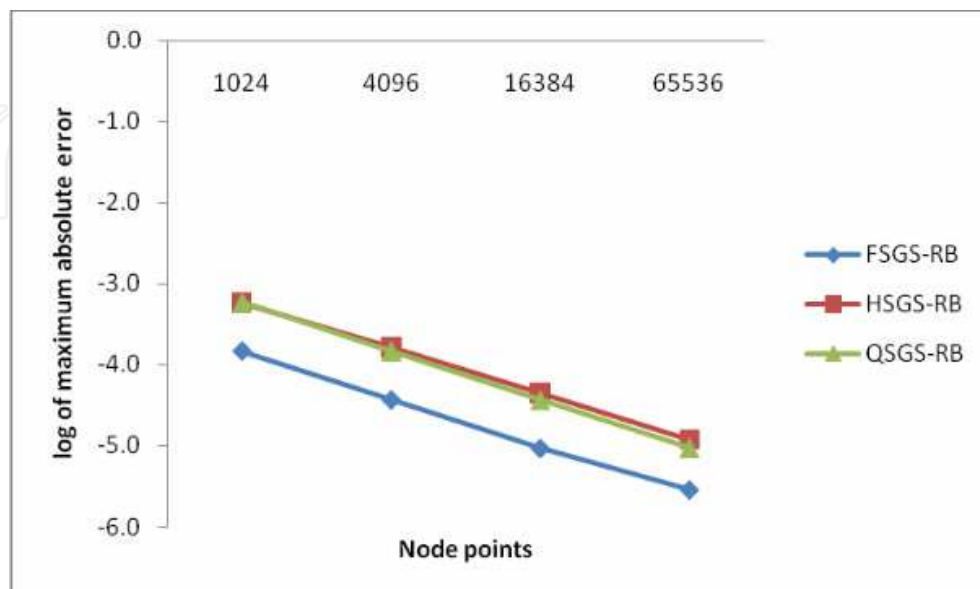


Fig. 14. Accuracy comparison for the full sweep, half sweep and quarter sweep schemes

7. Conclusion

In this chapter, we have demonstrated the development of three finite difference methods and three finite element methods. All methods utilised the red black ordering strategy and implementing the concept used in the Explicit Decoupled Group (EDG) and the Modified Explicit Group (MEG). By implementing the EDG, we developed a method called Half Sweep Successive Over-Relaxation utilising the Red Black ordering Strategy (HSSOR-RB) for finite difference scheme and Half Sweep Gauss-Seidel utilising the Red Black Strategy (HSGS-RB) for finite element scheme. Applying the concept used in MEG, we develop a method called Quarter Sweep Successive Over-Relaxation utilising the Red Black ordering strategy (QSSOR-RB) for finite difference scheme, while Quarter Sweep Gauss-Seidel utilising the Red Black ordering strategy (QSGS-RB) for finite element scheme. Both finite element schemes are developed using triangle element discretization.

The performance of both finite difference and finite element schemes are examined by comparing their number of iteration, computational time and accuracy to full sweep schemes, i.e. Full Sweep Successive Over-Relaxation utilising the Red Black Ordering Strategy (FSSOR-RB) for finite difference scheme and Full Sweep Gauss-Seidel utilising the Red Black Ordering Strategy (FSGS-RB) for finite element scheme. Helmholtz equation was used for testing the new finite difference scheme, while Poisson equation was used for testing the new finite element scheme.

From the numerical experiment, both HSSOR-RB and QSSOR-RB have shown the integrity to solve the Helmholtz equation faster than the FSSOR-RB. This is because The FSSOR-RB has the higher complexity and needs higher numbers of iteration than HSSOR-RB and QSSOR-RB methods. Having higher complexity and higher numbers of iteration makes the

method required the highest number of arithmetic operation compared to HSSOR-RB and QSSOR-RB. Furthermore, solving Helmholtz equation via QSSOR-RB only needs to solve a quarter of node points in the solution domain iteratively, while solving via HSSOR-RB only required to solve half of the node points in the solution domain. This is another reason why QSSOR-RB is faster than HSSOR-RB. The best part is HSSOR-RB and QSSOR-RB not only computes the Helmholtz problem faster than FSSOR-RB but also have similar accuracy to FSSOR-RB method.

Poisson equation has been used to examine the performance of the new finite element method, i.e. QSGS-RB and HSGS-RB. The numerical experiments show that HSGS-RB needs almost two times iteration number compared to QSGS-RB, while FSGS-RB needs almost four times iteration number compared to QSGS-RB. Besides, HSGS-RB only solve half of the solution domain iteratively, while QSGS-RB solve only quarter of the solution domain iteratively. Both are the reason why QSGS-RB are faster than HSGS-RB and FSGS-RB.

As the conclusion, applying the concept used in EDG and MEG with red black strategy produces fast solvers either using finite difference or finite element approaches.

8. Acknowledgement

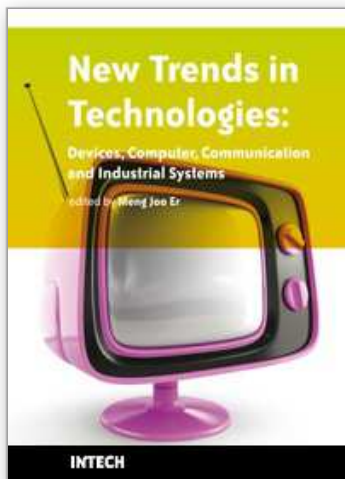
We acknowledge the financial support from 01-01-02-SF0677 (Malaysian Ministry of Science, Technology and Innovation Grant).

9. References

- Abdullah, A. R. (1991). The Four Point Explicit Decoupled Group (EDG) Method : A Fast Poisson Solver. *International Journal Computer Mathematics* Vol. (38) : 61-70.
- Belytschko, T.; Kringauz, Y. ; Organ, D. ; Fleming, M. & Krysl, P. (1996). Meshless Method : An Overview and Recent Developments. *Computer Methods in Applied Mechanics and Engineering* Vol. (139): 3-47.
- Evans, D. J. & Abdullah, A. R. (1982). Group Explicit Method for Parabolic Equations. *International Journal Computer Mathematics* Vol. (14) : 73-105.
- Evans, D. J. & Biggins, M. (1982). The Solution of Elliptic Partial Differential Equations by a New Block Over-Relaxation Technique. *International Journal Computer Mathematics* Vol. (10) : 269-282.
- Evans, D. J. & Yousif, W. F. (1990). The Explicit Block Relaxation Method as a Grid Smoother in the Multigrid V-cycle scheme. *International Journal Computer Mathematics* Vol. (34) : 71-78.
- Gupta, M. M. ; Kouatcou J. & Zhang, J. (1997). Comparison of Second and Fourth Order Discretizations for multigrid Poisson Solvers. *Journal of Computational Physics* Vol. (132 (2)) : 226-232.
- Ibrahim, A. (1993). The Study of the Iterative Solution of the Boundary Value problems by the Finite Difference Methods. PhD Thesis. Universiti Kebangsaan Malaysia.
- Ibrahim, A. & Abdullah, A.R. (1995). Solving the Two Dimensional Diffusion Equation by the Four Points Explicit Decoupled Group (EDG) Iterative Method. *International Journal Computer Mathematics* Vol. (58) : 253-263.
- Othman, M. & Abdullah, A.R. (2000). An Efficient Four Points Modified Explicit Group Poisson Solver. *International Journal Computer Mathematics* Vol. (76) : 203-217.

- Parter, S. V. (1998). Estimates for Multigrid Methods Based on Red Black Gauss-Seidel Smoothers. *Numerical Mathematics* Vol. (52) : 701-723.
- Rosser, J. B. (1975). Nine Points Difference Solution for Poisson's Equation. *Computers and Mathematics with Applications* Vol. (1) : 351-360.
- Yagawa, G. & Furukawa, T. (2000). Recent Developments of Free Mesh Method. *International Journal for Numerical Methods in Engineering* Vol. (47): 1419-1443.
- Yousif, W. S. & Evans, D.J. (1986a). Explicit Group Over-Relaxation Methods for Solving Elliptic Partial Differential Equations. *Mathematic and Computer in Simulation* Vol. (28) : 453-466.
- Yousif, W. S. & Evans, D.J. (1986b). Higher Order Difference Group Explicit (HODGE) Methods. *Communications in Applied Numerical Methods* Vol. (2) : 449-462.
- Zhang, J. (1996). Acceleration of Five Points Red Black Gauss-Seidel in Multigrid for Poisson Equations. *Applied Mathematics and Computation* Vol. (80(1)) : 71-78.
- Zhu, T. (1999). A New Meshless Regular Local Boundary Integral Equation (MRLBIE) approach. *International Journal for Numerical Methods in Engineering* Vol. (46): 123-125.

IntechOpen



New Trends in Technologies: Devices, Computer, Communication and Industrial Systems

Edited by Meng Joo Er

ISBN 978-953-307-212-8

Hard cover, 444 pages

Publisher Sciyo

Published online 02, November, 2010

Published in print edition November, 2010

The grandest accomplishments of engineering took place in the twentieth century. The widespread development and distribution of electricity and clean water, automobiles and airplanes, radio and television, spacecraft and lasers, antibiotics and medical imaging, computers and the Internet are just some of the highlights from a century in which engineering revolutionized and improved virtually every aspect of human life. In this book, the authors provide a glimpse of new trends in technologies pertaining to devices, computers, communications and industrial systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mohammad Khatim Hasan, Jumat Sulaiman, Samsul Ariffin Abdul Karim and Mohamed Othman (2010). Recent Development of Fast Numerical Solver for Elliptic Problem, *New Trends in Technologies: Devices, Computer, Communication and Industrial Systems*, Meng Joo Er (Ed.), ISBN: 978-953-307-212-8, InTech, Available from: <http://www.intechopen.com/books/new-trends-in-technologies--devices--computer--communication-and-industrial-systems/recent-development-of-fast-numerical-solver-for-elliptic-problem>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen