# Computational analysis of a mobile path-planning via Quarter-Sweep Two-Parameter Over-Relaxation

A'Qilah Ahmad Dahalan[1,2][0000-0003-2937-5204] and Azali Saudi[3][0000-0002-5025-4688]

[1] CONFIRM Centre for SMART Manufacturing, University of Limerick, Limerick, Ireland
[2] Centre for Defence Foundation Studies, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia
[3] Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia
A.Qilah@ul.ie

**Abstract.** Over the years, self-reliant navigation has risen to the forefront of research topics. Improving the path-planning competencies is an extremely important component in achieving excellent autonomous navigation. This paper describes a refinement of the proficiency of mobile path-planning through a computational approach, i.e., the Quarter-Sweep Two-Parameter Over-Relaxation (QSTOR), to solving path-planning problems iteratively. The solution of Laplace's equation (otherwise known as the harmonic functions) is the source for producing the potential function of the configuration space of the mobile robot. Numerical experiments illustrate that, in a given environment, a mobile robot is able to steer towards a particular destination with a smooth and ideal path from any beginning location. Furthermore, it is shown that in terms of the iterations number and computational time, the QSTOR iterative technique outperforms its predecessors in addressing mobile path-planning issues.

**Keywords:** Finite Difference Method, Accelerated Over-Relaxation, Optimal Route, Obstacle Avoidance, Quarter-Sweep Iterative Techniques.

## 1 Introduction

The robotics discipline is gaining traction in our daily lives as well as in various domains of modern industrial and cyber-physical automation. With the ability to embed intelligence into robots becoming more widely available, identifying the optimal solutions in the execution of any task, such as for path-planning and navigation, would be easily accomplished. These kinds of tasks could be said as one of the most complex challenges in intelligence robots. In the direction of constructing an autonomous mobile robot, it is important for the robot to be competent and accurate in creating a route as well as be collision-free. Practical algorithms concerning this difficulty have great exploitation such as in computer animation [1], robotics manufacturing [2], architectural design [3], including security, defence, and surveillance [4,5].

The aim of this paper is to use numerical potential functions on simulating a driving point-robot in the configuration space analogously by heat distribution [6]. The employment of such a heat transfer paradigm results in an environment with no local

minima, which give hugely beneficial for robot path-planning. Laplace's equation is utilized to depict the analogy of heat distribution across the experiments. The 'temperature values' for the path creation model in the environment, referred to as configuration space (C-space), are characterised by the solution of Laplace's equation, i.e., the harmonic functions. To solve these functions, a variety of approaches have been explored, while numerical techniques are most typically used due to their fast-processing mechanisms and proficiency in solving the problem. This paper conducted a number of tests to examine the performance of proposed accelerated algorithms in generating mobile robot paths.

## 2 Path-planning Structure

Path-planning, in general, allows an autonomous vehicle or a robot to discover the shortest and safest most obstacle-free path from a starting point to a destination. Indoor mobile robot path navigation can be achieved in many different ways. A path navigation algorithm for an identified environment can certainly yield a series of nodes for a robot to trail. Typically, a grid of a predetermined size is created to evaluate different algorithms, showing where "passable" is on the C-space. It is reasonable to assume that the robot can traverse all of the grid's boundaries.

The structure of this experiment is based on the use of a point-robot to simulate the motion within the recognized C-space. The robot's route is determined using a heat transfer analogy in which the target point (with the lowest potential value) serves as a heat-pulling sink. While every wall and obstacle (with the highest potential value) is regarded as a heat source that should always be set as constant. In compliance with the heat transfer behavior, the heat will flow from a higher-temperature region towards a lower-temperature region, completing the C-space. This event is represented by harmonic function values, which will result in so-called heat flux lines flowing/streaming towards the region with the lowest potential value, i.e., the sink. The path line for the robot to traverse across the C-space was built out in this arrangement, by following the heat flux line produced. The implementation of the harmonic function prevents the event of local minima and can guide the robot to avoid obstacles in the environment [7].

### 2.1 Harmonic Functions

A Laplace's equation-satisfying function is known as a harmonic function provided in the domain $\Omega \subset R^n$. The borderline of every wall, each obstacle in the region, primary points, and target points are all contained within the boundary of $\Omega$ for the development of the robot path. Consider Laplace's equation below with $x_i$ is the i$^{th}$ coordinates in the Cartesian plane, and $n$ is the dimension.

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0 . \tag{1}$$

By using the numerical approach, i.e., Jacobi or Gauss-Seidel (GS), Laplace's equation (1) could be adequately solved. The harmonic function has been shown that it abides by the min-max principle, which implies it prevents the formation of spurious local minima excluding the target point and typically creates a smooth path [8]. For this reason, the harmonic potential technique is a viable and appealing decision for robot path-planning. Most often, conventional methods [9-11] are used to solve the Laplace equation. Equation (1) in this paper was solved using the quarter-sweep iterative approach to improve the acceleration of the computational execution.

A global approach is used to measure the harmonic potentials of the robot C-space for path-planning problems. The trail lines for a robot to move along from start to end location without encountering any obstacles are mapped using potential solutions for equation (1). As mentioned earlier, obstacles and walls are viewed as current sources whiles the target point is to be the sink. The Dirichlet boundary conditions provide boundary values. Following that, by performing a standard Gradient Descent Search (GDS) on the potential field, a sequence of potential points with lower values is found, progressing to the point with the lowest potential value, which is the target location.

Altogether, this paper attempt to replicate the stated path-planning paradigm, defining the solution of Laplace's equation over the resemblance of temperature (for the potential) and heat flow (for the path line). The experimentation takes place on a two-dimensional domain with assorted shapes of obstacles, along with the walls. To address equation (1) in gaining potential values for each node, the Quarter-Sweep Two-Parameter Over-Relaxation (QSTOR) scheme is employed. The existing technique (i.e., families of over-relaxation methods) were also measured for comparison to analyse the competence of the proposed scheme.

## 3 Materials and Techniques

From equation (1), the two-dimensional Laplace's equation is given as

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 . \tag{2}$$

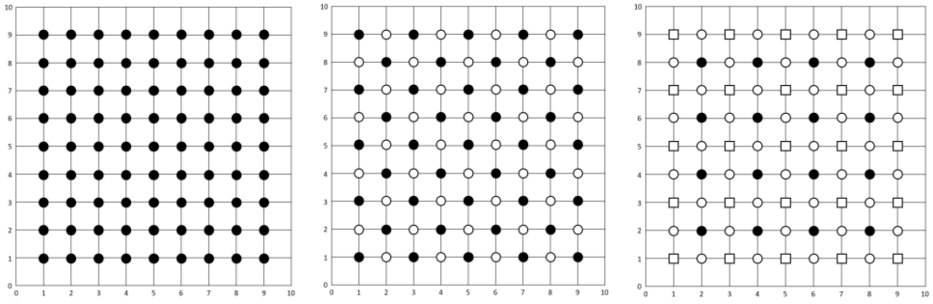
The Laplacian operator is implied by $\nabla^2$. To compute equation (2) using a numerical method, it should be discretized over the simplest five-point finite difference approximation (5P-FDA). For two-dimensional Laplace's equation (2), let $U_{i,j}$ approaches the solution of $u$ along the grid point $\left(x_i, y_j\right)$, hence the discretization of these Laplace equations by conventional five-point stencil is written as

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j} = 0 . \tag{3}$$

The iterative routine for Laplace's equation (2) is implying swapping the node value continuously with the median of its four neighbours. In parallel, all nodes in the grid

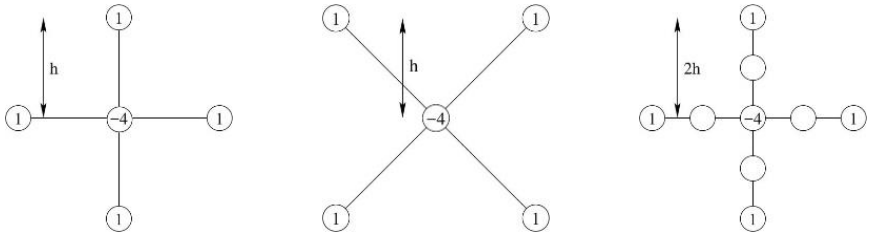
point will be computed using equation (3), this action is called full-sweep (FS) iteration (see Fig. 1(a)). Abdullah [12] later initiated the Explicit Decoupled Group, which was then known as the half-sweep (HS) approach. This method demonstrates an effective technique for solving PDEs [13-16]. Since the HS technique yielded such promising results, Othman and Abdullah [17] came out with an improved approach namely Modified Explicit Group, also known as quarter-sweep (QS). Fig. 1 indicates the computational mesh of each sweep technique, where only black points are evaluated for the whole iteration cycle. In the mesh region, only half and a quarter of the node points are calculated using HS and QS schemes, respectively. Rationally, this signifies the reduction of computational time on each iteration. Fig. 2 shows the computational stencils of each technique. It is observed that the HS iteration is primarily based on rotated 5P-FDA in solving the Laplace equation, given as

$$U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - 4U_{i,j} = 0\,. \tag{4}$$



(a) (b) (c)

**Fig. 1.** The computational mesh of (a) FS, (b) HS, and (c) QS technique.




(a) (b) (c)

**Fig. 2.** The computational stencil of (a) FS, (b) HS, and (c) QS technique.

### 3.1 Conceptualization of the QS Method

The implementation of the QS iterative scheme will compute only 1 out of 4 of the nodal points at one time (see Fig. 1(c)) during the iteration process in the C-space. Consequently, it will decrease the computational complexity drastically i.e., roughly 75%. The QS approximation equation precisely skipped two nodal points from the

mesh space (see Fig. 2(c)). Therefore, the formula of QS five-point approximation be written as

$$U_{i-2,j}+U_{i+2,j}+U_{i,j-2}+U_{i,j+2}-4U_{i,j}=0. \tag{5a}$$

Considering finite difference from equation (5a), the GS iterative technique for QS can be rewritten and denoted as

$$U_{i,j}^{(k+1)}=\frac{1}{4}\left[U_{i-2,j}^{(k+1)}+U_{i+2,j}^{(k)}+U_{i,j-2}^{(k+1)}+U_{i,j+2}^{(k)}\right]. \tag{5b}$$

Successive Over-Relaxation (SOR) is basically a variant of the GS technique. When implanted SOR approach into equation (5) by appending a weighted parameter $\omega$ [18], the QSSOR iterative scheme is given as

$$U_{i,j}^{(k+1)}=\frac{\omega}{4}\left[U_{i-2,j}^{(k+1)}+U_{i+2,j}^{(k)}+U_{i,j-2}^{(k+1)}+U_{i,j+2}^{(k)}\right]+(1-\omega)U_{i,j}^{(k)}. \tag{6}$$

Be noted that whenever $\omega=1$, then the SOR approach is in fact simplified to the GS method.

The Accelerated Over-Relaxation (AOR) fundamentally is a simplification of the SOR technique with additional optimal parameters, denoted as $\omega$ and $\omega'$ in this paper. To execute the AOR scheme as proposed in [19], the node points of $u_{i-1,j-1}^{(k+1)}$ and $u_{i+1,j-1}^{(k+1)}$ are interchanged to $u_{i-1,j-1}^{(k)}$ and $u_{i+1,j-1}^{(k)}$ respectively, as well as inserting the $\dfrac{\omega'\left(u_{i-1,j-1}^{(k+1)}-u_{i-1,j-1}^{(k)}\right)}{4}$ and $\dfrac{\omega'\left(u_{i+1,j-1}^{(k+1)}-u_{i+1,j-1}^{(k)}\right)}{4}$ nodes into equation (6). Now, the new scheme of QSAOR is provided as

$$\begin{aligned}U_{i,j}^{(k+1)}=&\frac{\omega'}{4}\left[U_{i-2,j}^{(k+1)}-U_{i-2,j}^{(k)}+U_{i,j-2}^{(k+1)}-U_{i,j-2}^{(k)}\right]\\&+\frac{\omega}{4}\left[U_{i-2,j}^{(k)}+U_{i+2,j}^{(k)}+U_{i,j-2}^{(k)}+U_{i,j+2}^{(k)}\right]+(1-\omega)U_{i,j}^{(k)}\end{aligned}. \tag{7}$$

Meanwhile, the Two-parameter Over-Relaxation (TOR) technique is indeed a deduction from the AOR scheme. The main intention of this technique is to improve the convergence speed, ergo of it consists three different relaxation parameters, $\omega$, $\omega'$, and $\omega''$. Thus, the QSTOR iterative scheme is

$$\begin{aligned}U_{i,j}^{(k+1)}=&\frac{\omega'}{4}U_{i,j-2}^{(k+1)}+\frac{\omega''}{4}U_{i-2,j}^{(k+1)}+\frac{\omega}{4}\left(U_{i,j+2}^{(k)}+U_{i+2,j}^{(k)}\right)\\&+\left(\frac{\omega-\omega'}{4}\right)U_{i,j-2}^{(k)}+\left(\frac{\omega-\omega''}{4}\right)U_{i-2,j}^{(k)}+(1-\omega)U_{i,j}^{(k)}\end{aligned}. \tag{8}$$

The uncertainty of relaxation parameter values has resulted in the minimum iteration counts. Previous researchers [19,20] specified that the values of $\omega'$ and $\omega''$ are generally chosen remain near to the SOR $\omega$ value. The computation is then recurrent for a range of $1 \le \omega < 2$. So as to discover the optimum value, the relaxation parameter values are individual for each sweep case, as certain values are not converged in some cases. Additionally, as the values of each parameter are predetermined before execution, the impact of complexity on determining the value of parameters on the entire computation is unaffected. It will certainly shift if the few ranges of parameter values are set in the computation algorithm. The implementation of the QSTOR scheme to solve Laplace's problem (2) is described in Algorithm 1.

**Algorithm 1. QSTOR iterative scheme**

i. Set up the C-space through the designated start and target points.

ii. Initialising starting point $U, \varepsilon \leftarrow 10^{-15}, iteration \leftarrow 0$.
For every • node points, calculate

iii. $$U_{i,j}^{(k+1)} \leftarrow \frac{\omega'}{4} U_{i,j-2}^{(k+1)} + \frac{\omega''}{4} U_{i-2,j}^{(k+1)} + \frac{\omega}{4}\left(U_{i,j+2}^{(k)} + U_{i+2,j}^{(k)}\right) + \left(\frac{\omega-\omega'}{4}\right) U_{i,j-2}^{(k)} + \left(\frac{\omega-\omega''}{4}\right) U_{i-2,j}^{(k)} + (1-\omega) U_{i,j}^{(k)}.$$

Compute the remaining □ node points via the direct method

iv. $$U_{i,j}^{(k+1)} \leftarrow \frac{1}{4}\left[U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right],$$

and ○ node points by using

$$U_{i,j}^{(k+1)} \leftarrow \frac{1}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k)}\right].$$

v. Verify the convergence test for $\varepsilon \leftarrow 10^{-15}$, then perform GDS to create a path towards the target. Otherwise, go back to step (iii).

## 4 Experiments and Results

There are four different C-spaces (with assorted obstacles) over four separate mesh sizes through the simulation experiments in this study. Although no specific potential values were appointed to any starting position, the target point was placed at the lowermost temperature values. During the initial setting, every obstacle and wall were assigned with the highest potential value where boundary values are described by the Dirichlet boundary conditions. The free spaces in the environment were made to be zero potential.

The computational process was carried out using an AMD A10-7400P Radeon R6 with 10 Compute Cores 4C+6G running at 2.50GHz and 8GB of RAM. Provided that the state for stopping criteria is satisfied, the process of iteratively measuring potential

values at each point continues. The iteration loop will be terminated, where the variance of the computational values was extremely small (i.e., $1.0^{-15}$), if the potential values do not show any further changes. This level of precision was necessary for the solutions to avoid saddle points, which are flat areas that fail to produce routes.

The iteration number and the execution time for every computational approach is respectively shown in Table 1 and 2. As compared to other suggested techniques, the QSTOR iterative scheme has been proven that it is significantly faster. It is demonstrated that, in terms of iteration number, the QSTOR outperformed the QSAOR (approximately by 5% to 12%) and QSSOR (approximately by 15% to 28%). On the other hand, the QSTOR decreases QSSOR from 10% to 18% and QSAOR from 9% to 20% in terms of execution time.

**Table 1.** Findings of the proposed schemes for **iteration number**.

| | **Techniques** | **N x N** | | | |
|---|---|---|---|---|---|
| | | ***300*** | ***600*** | ***900*** | ***1200*** |
| Condition 1 | FSSOR | 1728 | 8117 | 17831 | 31346 |
| | FSAOR | 1591 | 7529 | 16594 | 28984 |
| | FSTOR | 1656 | 7815 | 17199 | 27895 |
| | HSSOR | 837 | 4108 | 9086 | 15892 |
| | HSAOR | 759 | 3803 | 8420 | 14768 |
| | HSTOR | 797 | 3949 | 8721 | 14234 |
| | QSSOR | 351 | 2078 | 4632 | 8113 |
| | QSAOR | 348 | 1913 | 4280 | 7508 |
| | QSTOR | 344 | 1992 | 4448 | 7279 |
| Condition 2 | FSSOR | 2228 | 8776 | 19254 | 33558 |
| | FSAOR | 2006 | 7973 | 17538 | 30573 |
| | FSTOR | 1893 | 7553 | 16642 | 29008 |
| | HSSOR | 1071 | 4438 | 9813 | 17149 |
| | HSAOR | 944 | 4023 | 8924 | 15614 |
| | HSTOR | 877 | 3811 | 8461 | 14813 |
| | QSSOR | 452 | 2229 | 5014 | 8771 |
| | QSAOR | 430 | 2007 | 4542 | 7976 |
| | QSTOR | 414 | 1890 | 4305 | 7558 |
| Condition 3 | FSSOR | 3624 | 14644 | 33004 | 57484 |
| | FSAOR | 3236 | 13165 | 29680 | 51738 |
| | FSTOR | 2843 | 11685 | 26393 | 46021 |
| | HSSOR | 1780 | 7445 | 16856 | 29418 |
| | HSAOR | 1568 | 6681 | 15149 | 26456 |
| | HSTOR | 1349 | 5909 | 13463 | 23523 |
| | QSSOR | 828 | 3769 | 8624 | 15061 |
| | QSAOR | 698 | 3366 | 7740 | 13545 |
| | QSTOR | 512 | 2960 | 6856 | 12023 |
| Condition 4 | FSSOR | 2507 | 9868 | 21654 | 37762 |
| | FSAOR | 2288 | 9025 | 19840 | 34601 |
| | FSTOR | 2067 | 8217 | 18052 | 31519 |
| | HSSOR | 1212 | 5000 | 11036 | 19288 |
| | HSAOR | 1097 | 4555 | 10098 | 17670 |
| | HSTOR | 967 | 4141 | 9180 | 16085 |
| | QSSOR | 555 | 2502 | 5638 | 9873 |
| | QSAOR | 467 | 2287 | 5148 | 9030 |
| | QSTOR | 427 | 2066 | 4676 | 8215 |

**Table 2.** Findings of the proposed schemes for the **execution time** (in second).

| | Techniques | N x N | | | |
|---|---|---|---|---|---|
| | | ***300*** | ***600*** | ***900*** | ***1200*** |
| Condition 1 | FSSOR | 8.13 | 227.95 | 1134.25 | 3728.92 |
| | FSAOR | 8.61 | 230.17 | 1148.87 | 3692.74 |
| | FSTOR | 7.60 | 233.91 | 1188.08 | 3565.09 |
| | HSSOR | 2.39 | 81.24 | 404.15 | 1375.27 |
| | HSAOR | 1.72 | 73.76 | 369.91 | 1247.65 |
| | HSTOR | 2.55 | 84.84 | 413.84 | 1335.52 |
| | QSSOR | 0.39 | 14.99 | 81.55 | 293.92 |
| | QSAOR | 0.56 | 15.83 | 84.47 | 292.46 |
| | QSTOR | 0.38 | 16.46 | 87.40 | 279.95 |
| Condition 2 | FSSOR | 10.69 | 251.72 | 1270.23 | 4077.22 |
| | FSAOR | 10.27 | 248.24 | 1226.66 | 3976.33 |
| | FSTOR | 9.39 | 233.83 | 1194.50 | 3732.02 |
| | HSSOR | 2.95 | 86.77 | 445.70 | 1423.27 |
| | HSAOR | 2.75 | 76.79 | 403.25 | 1263.63 |
| | HSTOR | 2.70 | 82.42 | 401.42 | 1326.65 |
| | QSSOR | 0.64 | 16.69 | 90.03 | 313.44 |
| | QSAOR | 0.56 | 16.68 | 89.98 | 314.14 |
| | QSTOR | 0.52 | 15.19 | 85.08 | 287.87 |
| Condition 3 | FSSOR | 16.22 | 427.27 | 2190.45 | 7432.68 |
| | FSAOR | 18.66 | 418.45 | 2073.25 | 7254.02 |
| | FSTOR | 15.20 | 369.55 | 1927.30 | 6300.13 |
| | HSSOR | 5.16 | 154.79 | 783.72 | 2634.52 |
| | HSAOR | 4.80 | 137.18 | 721.94 | 2300.84 |
| | HSTOR | 4.30 | 135.81 | 661.90 | 2262.25 |
| | QSSOR | 0.92 | 30.04 | 166.12 | 567.28 |
| | QSAOR | 1.08 | 29.24 | 161.76 | 570.33 |
| | QSTOR | 0.77 | 25.35 | 144.71 | 488.66 |
| Condition 4 | FSSOR | 11.02 | 281.85 | 1441.47 | 4853.57 |
| | FSAOR | 12.52 | 281.78 | 1423.54 | 4743.21 |
| | FSTOR | 10.91 | 255.82 | 1292.23 | 4269.42 |
| | HSSOR | 3.58 | 102.16 | 510.22 | 1686.65 |
| | HSAOR | 3.08 | 92.44 | 471.17 | 1511.93 |
| | HSTOR | 2.99 | 93.87 | 458.45 | 1527.54 |
| | QSSOR | 0.75 | 19.85 | 106.87 | 369.38 |
| | QSAOR | 0.73 | 19.97 | 108.78 | 364.51 |
| | QSTOR | 0.66 | 17.80 | 94.22 | 320.61 |

### 4.1 Discussion

The moment the potential values were gained, the route was constructed by carrying out the steepest descent search following the initial points to the specified destination. The development of path creation was brief, wherein the algorithm plainly picks the lowest temperature value of its adjacent points from the current point. This action remains until the marked target point is achieved. In accordance with the heat transfer analogy with numerical computation, the paths were favourably generated in an obstacle environment as shown in Fig. 3. Each and every single beginning point (green point) successfully reached the designated destination position (red point) and evaded various obstacles set in the C-space. Through Robot 2D Simulator [21], the simulations solely evaluate known static two-dimensional indoor configurations.

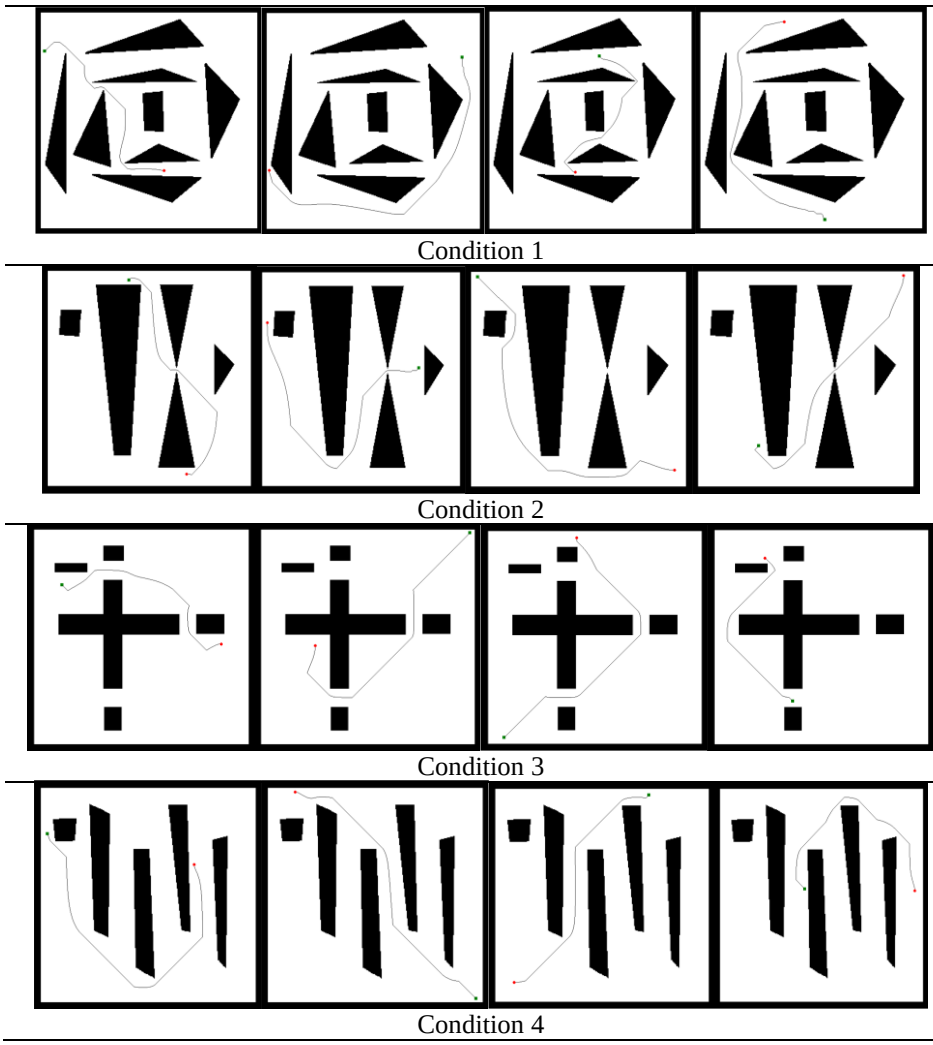



**Fig. 3.** The produced pathways from various start (green point) and goal (red point) points for varied C-space.

To simplify the data, the line graph of the iteration counts and the time taken for every condition was presented in Fig. 4 and Fig. 5 respectively. Clearly shows that all four conditions provide a similar pattern, demonstrating that the QSTOR scheme produced the best outcomes in developing and completing the path as compared to other techniques for both iteration counts as well as CPU time. It can be deduced from the results table and the line chart that utilizing the HS approach has resulted in a nearly and more than 50% reduction than using the standard procedure. Whereas, nearly 75% diminution has taken from QS technique as against conventional technique.

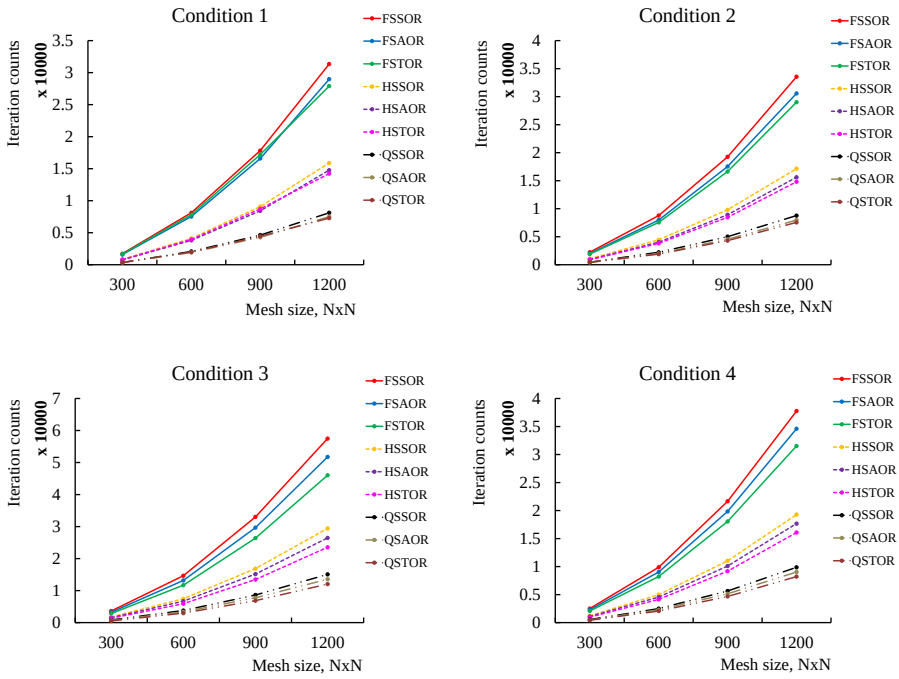



**Fig. 4.** The performance graph concerning the iteration counts in various C-space sizes.

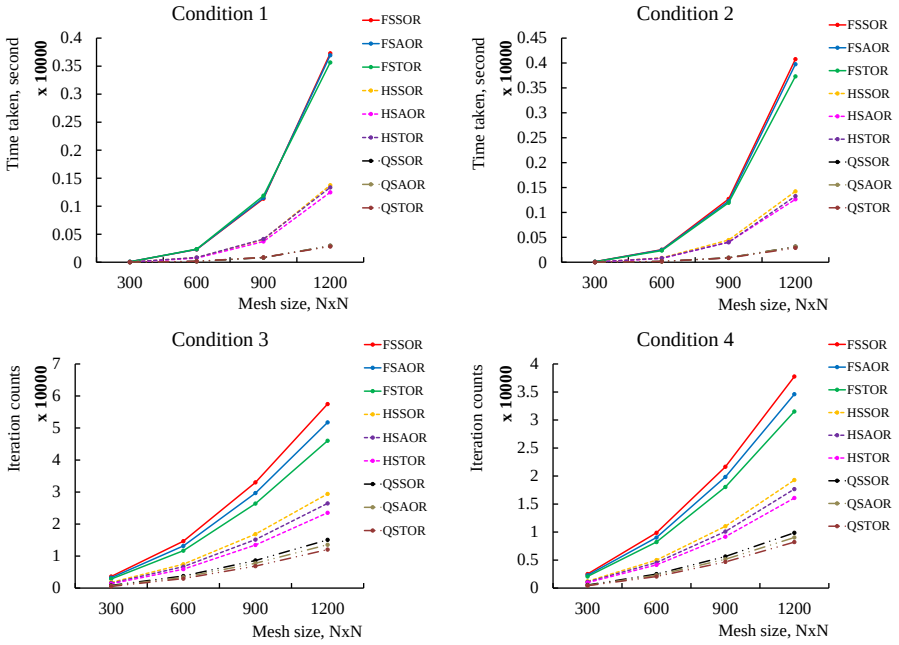



**Fig. 5.** The performance graph concerning the time taken in various C-space sizes.

Concerning the computational complexity analysis of all iterative methods considered, it is assumed that each arithmetic operation requires one unit of computational time. Theoretically, as the complexity analysis is reduced, the number of iterations will become lesser thus decreasing the CPU time. Even though the number of arithmetic operations for the families of the TOR method is more compared to families of SOR as well as AOR, they converge faster since the presence of weighted parameters [22]. The remaining points, on the other hand, will be omitted in the whole calculation of the computational complexity since they will give no significance to the computation as it does not contribute to the changes in the calculation. After all, the loop for the remaining point is only at one.

It is obvious that the computational complexities of the FS algorithms are reduced drastically by the HS and QS algorithms by approximately 50% and 75%, respectively. As discussed before, only half of the node points are involved during the iteration process of the HS algorithms. For QS algorithms, the iteration process only involves a quarter of node points. Therefore, by reducing the amount of node points involved during the iteration process, convergence can be achieved much faster, thus improving the overall performance of the iterative methods and the path searching process. As for the relation between computational complexities and CPU time, it shows that the higher the complexity, often resulting in higher CPU time.

## 5 Conclusions

Owing to the fact the recently developed and newly found techniques, along with the availability of fast machines today, this experiment demonstrates that the solution to mobile path-planning problems through numerical approaches is, in fact, creative and doable. The results table shows that the TOR iterative scheme, in contrast to conventional SOR and AOR techniques, was faster in terms of iteration counts and processing time. The results are unaffected by an increasing number of obstacles because the computing process is only becoming faster as the calculation ignores or disregards the zones occupied by the obstacles. The edge of the proposed algorithm is that it allows the robot to move from starting position to the ending position safely along the shortest path, regardless of the obstacles' size, form, or placement.

## Acknowledgment

This research was financially supported by Universiti Pertahanan Nasional Malaysia. The authors also acknowledge support from Science Foundation Ireland (SFI) under Grant Number SFI/16/RC/3918 (Confirm), and Marie Skłodowska-Curie grant agreement No. 847577 co-funded by the European Regional Development Fund.

## References


1. Ye, Y., Song, Z., Zhao, J.: High-fidelity 3D real-time facial animation using infrared structured light sensing system. Computer & Graphics 104, 46-58 (2022).
2. Liu, Z., Liu, Q., Xu, W., Wang, L., Zhou, Z.: Robot learning towards smart robotic manufacturing: A Review. Robotics and Computer-Integrated Manufacturing 77, 102360 (2022).
3. Soliman, M., Avgeriou, P., Li, Y.: Architectural design decisions that incur technical debt-An industrial case study. Information and Software Technology 139, 106669 (2021).
4. Ahmad, R.W., Hasan, H., Yaqoob, I., Salah, K., Jayaraman, R., Omar, M.: Blockchain for aerospace and defense: Opportunities and open research challenges. Computers & Industrial Engineering 151, 106982 (2021).
5. Becker, M., Faucher, P.: Recent developments in the implementation of European Space Surveillance and Tracking (EU SST)-Security and data policy. Journal of Space Safety Engineering 8(2), 178-181 (2021).
6. Dahalan, A.A., Saudi, A., Sulaiman, J., Din, W.R.W.: Robot navigation in static indoor environment via accelerated iterative method. Advanced Science Letters 24(2), 986-989 (2018).
7. Connolly, C.I., Burns, J.B., Weiss, R.: Path planning using Laplace's equation. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2102-2106. Cincinnati, OH (1990).
8. Connolly, C.I., Gruppen, R.: On the applications of harmonic functions to robotics. Journal of Robotics System 10(7), 931-946 (1993).
9. Al-Khaled, K.: Numerical solutions of the Laplace's equation. Applied Mathematics and Computation 170(2), 1271-1283 (2005).
10. Shivaram, K.T., Jyothi, H.R.: Finite element approach for numerical integration over family of eight node linear quadrilateral element for solving Laplace equation. Materials Today:Proceedings 46(9), 4336-4340 (2021).
11. Liu, Y.C., Fan, C.M., Yeih, W., Ku, C.Y., Chu, C.L.: Numerical solutions of two-dimensional Laplace and biharmonic equations by the localized Trefftz method. Computers and Mathematics with Application 88, 120-134 (2021).
12. Abdullah, A.R.: The four point explicit decoupled group (EDG) method: A fast Poisson solver. International Journal of Computer Mathematics 38(1-2), 61-70 (1990).
13. Abdullah, A.R., Ali, N.H.M.: A comparative study of parallel strategies for the solution of elliptic PDEs. Parallel Algorithms and Applications 10, 93-103 (1996).
14. Sulaiman, J., Hassan, M.K., Othman, M.: The half-sweep iterative alternating decomposition explicit (HSIADE) method for diffusion equation. In: Zhang, J., He, J.H., Fu, Y. (eds.) Computational and Information Science, LNCS, vol. 3314, pp. 57-63. Springer, Berlin, Heidelberg (2004).
15. Dahalan, A.A., Saudi, A., Sulaiman, J., Din, W.R.W.: Autonomous navigation in static indoor environment via rotated Laplacian operator. In: AIP Conference Proceedings 1974, 0200352018, (2018).
16. Dahalan, A.A., Saudi, A.: Rotated TOR-5P Laplacian iteration path navigation for obstacle avoidance in stationary indoor simulation. In: iCITES2020, Advances in Robotics, Automation and Data Analytics, Advances in Intelligent Systems and Computing, vol. 1350, pp. 285-295. Springer Cham (2021).
17. Othman, M., Abdullah, A.R.: An efficient four points modified explicit group Poisson solver. International Journal Computer Mathematics 76, 203-217 (2000).

18. Young, D.M.: Iterative methods for solving partial difference equations of elliptic type. Transactions of the American Mathematical Society 76, 92-111 (1954).
19. Ali, N.H.M., K. Pin, F.: Modified explicit group AOR methods in the solution of elliptic equations. Applied Mathematical Sciences 6(50), 2465-2480 (2012).
20. Hadjidimos, A.: Accelerated overrelaxation method. Mathematic Computation 32(141), 149-157 (1978).
21. Saudi, A.: Robot Path Planning using Family of SOR Iterative Methods with Laplacian Behaviour-Based Control, PhD thesis, Universiti Malaysia Sabah, Malaysia (2015).
22. Kuang, J., Ji, J.: A survey of AOR and TOR methods. Journal of Computational and Applied Mathematics 24, 3-12 (1988).