

Indoor Path Planning for Mobile Robot using Half-Sweep Gauss-Seidel via Nine-Point Laplacian (HSGS9L)

Azali Saudi¹ and Jumat Sulaiman²

¹*School of Engineering and Information Technology*

²*School of Science and Technology*

University of Malaysia Sabah

Kota Kinabalu, MALAYSIA

Abstract

This paper proposed a numerical computation for solving path planning problem for a mobile robot operating in indoor environment grid model. It is based on the use of Laplace's Equation to constraint the distribution of potential values in the configuration space of a mobile robot. Consequently, the solution of Laplace's Equation is computed by employing Half-Sweep Gauss-Seidel via Nine-Point Laplacian (HSGS9L) iterative method. The simulation results show that this half-sweep iteration performs much faster than the previous methods in generating smooth path for mobile robot to move from start to goal position.

1. Introduction

In industrial automation, the robot is often required to have the capability of moving quickly from a given initial point to a specified goal point. During its exploration, the robot must be able to avoid from colliding with any obstacles in its environment model. Thus, path planning problem is considered one of the most important issue in constructing a truly autonomous vehicle. This study attempts to provide the solution to the path planning problem for a mobile robot operating in indoor environment model by using a global numerical approach. It is inspired by the theory of heat transfer analogy. Based on this heat transfer model, the configuration space is modeled with Laplace's equation. Consequently, the solutions of Laplace's equation also called harmonic functions that represent temperature values distribution in the configuration space will be used to simulate the generation of path for mobile robot motion. In the past, various approaches had been used to obtain harmonic functions, but the most common method is

via numerical techniques due to the availability of fast processing machine and their elegant and efficiency in solving the problem. In this work, we investigate the performance of Gauss-Seidel with half-sweep iteration method based on 9-point formula, better known as Half-Sweep Gauss via Nine-Point Laplacian (HSGS9L) iterative method to obtain the harmonic functions for generating mobile robot path in varying sizes of environment model.

2. Literature Review

Pioneer work by Connolly and Gruppen [1] shows that harmonic functions have a number of properties useful in robotic applications. Khatib [2] utilized the use of potential functions for robot path planning, in which every obstacle produces a repelling force and the goal exerts an attractive force. In contrast, the work by Koditschek [3] concluded that geometrically at least in certain types of domains, the potential functions can be used to guide the effector from almost any point to a given point. All of these potential field methods, however, suffer from the generation of local minima. Connolly *et al.* [4] and Akishita *et al.* [5], both of them developed independently a global method that generates smooth path by using solutions to Laplace's equations, where the potential fields were computed in a global manner over the entire region. Meanwhile, Waydo & Murray [6] studied the use of stream functions for vehicle motion. More recently, Szulczyński *et al.* [7] employed harmonic functions for real-time obstacle avoidance.

Various approaches had been used to obtain harmonic functions in the past. The standard are Jacobi and Gauss-Seidel [8] point iterative methods. Daily and Bevely [9] employed analytical solution for arbitrarily shaped obstacles. Garrido *et al.* [10] computed the harmonic functions with finite elements for robotic motion. Then, Abdullah [11]

introduced half-sweep iteration for his works on Explicit Decoupled Group (EDG) iterative method for solving 2-D Poisson equations. This half-sweep iterative method was also applied in solving partial differential equations in Ibrahim & Abdullah [12], Yousif & Evans [13], and Abdullah & Ali [14]. A modified version of this method was also investigated by Sulaiman *et al.* for solving diffusion equation [15].

3. Configuration Space Model

Mobile robot path planning problem can be modeled as a well-known steady-state heat transfer problem, where heat sources come from the boundaries and the heat sink will pull the heat in. This heat conduction process produces a temperature distribution and the heat flux lines that are flowing to the sink fill the workspace. In a mobile robot environment setup, the goal point is treated as a heat sink whilst the boundary walls and obstacles are considered as heat sources that are fixed with constant temperature values. Once the temperature distribution in the field is obtained, it will be used as a guide to generate path for mobile robot to move from the start point to the goal point. The idea is to follow the heat flux that will flow from high temperature sources to the lowest temperature point in the environment. The temperature distribution of the configuration space is computed by employing harmonic function to model the environment setup.

Mathematically, a harmonic function on a domain $\Omega \subset R^n$ is a function which satisfies Laplace's equation, in which x_i is the i -th Cartesian coordinate, and n is the dimension. In the case of robot path construction, the domain Ω consists of the outer boundary walls, all obstacles in the workspace, start points and the goal point.

$$\nabla^2 \phi = \sum_{i=1}^n \frac{\partial^2 \phi}{\partial x_i^2} = 0 \quad (1)$$

The equation of Eq. (1) can be solved efficiently using numerical method. Standard methods are point Jacobi and Gauss-Seidel iterative method. In this study, the robot model is represented by a point in the configuration space. The configuration space is designed in grid form. The function values associated with each node are then computed iteratively via numerical technique to satisfy equation in Eq. (1). The highest temperature is assigned to the start point whereas the goal point is assigned the lowest, meanwhile different initial temperature values are assigned to the outer wall boundaries and obstacles. Initial temperature values are not required to be assigned to the start points.

The solutions to the Laplace's Equation are examined with Dirichlet boundary conditions.

4. Formulation of Half-Sweep Gauss-Seidel via Nine-Point Laplacian

Our previous works that employed a block iterative method using 9-point formula (also known as 9-point Laplacian) [19, 20] performed much faster than the iteration using standard 5-point formula [16, 17], except for half-sweep iteration [18] which is essentially computed only half of the total nodes in the configuration space. In this study, we propose an improved version of [18] by employing 9-point formula into the equation. The proposed method is now known as Half-Sweep Gauss-Seidel via Nine-Point Laplacian (HSGS9L) would iterate only half of the node points in the configuration space but include 8 neighbouring points in its formulation. By adding more points in the formulation, the accuracy of each node point calculation get higher, thus leads to faster convergence rate. With HSGS9L iterative method, the computation involves node of black points only, thus in contrast to half-sweep iteration as shown in Figure 1 (a), the half-sweep iteration of HSGS9L as shown in Figure 1 (b) involves only half of the whole node points.

Essentially, the standard full-sweep iteration uses 5-point stencil shown in Figure 2 (a), whereas half-sweep iteration is actually based on the five-point 45-degree rotated finite difference approximation as shown in Figure 2 (b). The main characteristic of half-sweep iteration is the reduction of computational complexity by considering only half of the total node points.

In the case of 9-point formulation, full-sweep iteration uses stencils shown in Figure 3 (a), whereas in the implementation of half-sweep iteration with HSGS9L the stencil shown in Figure 3 (b) is used. Figure 4 (a) shows the five black points involve in each calculation of standard 5-point half-sweep iteration, whereas Figure 4 (b) shows the nine black points to be considered in each calculation of HSGS9L iterative method. Now, let us consider the two-dimensional Laplace's equation in Eq. (1) defined as

$$\frac{\partial^2 U}{\partial^2 x} + \frac{\partial^2 U}{\partial^2 y} = 0 \quad (2)$$

Then, the discretization of Eq. (2) based on the Nine-Point Laplacian can be defined as below

$$4(U_{i+1,j-1} + U_{i+1,j+1} + U_{i-1,j+1} + U_{i-1,j-1}) + U_{i+2,j} + U_{i,j+2} + U_{i-2,j} + U_{i,j-2} - 20U_{i,j} = 0 \quad (3)$$

The 9-point finite difference approximation equation in Eq. (3) is then used to generate linear

system. To solve the generated linear system, the iterative process is run until the maximum error falls into a specified tolerance error, in which the iteration stops. It is important that tolerance error is set to a very small value since high precision is essential in order to avoid (or at least minimize) the occurrence of flat area in the final solution. Once the temperature values of all black points are obtained, approximate values of the remaining white points will be obtained directly by standard direct method computation in single iteration.

5. Experiments and Results

The experiment was carried out with varying sizes of static environment model, i.e. 128x128, 256x256 and 512x512, that consists of a goal point, three starting points and varying setup of inner walls and outer boundary walls. Initially, the inner and outer walls were fixed with high temperature values, whereas the goal point was set to very low temperature, and all other free spaces were set to zero temperature value. The experiments run on Intel Core 2 Duo CPU running at 3 GHz speed with 1GB of RAM. The software code, now known as RobotPath Simulator, was written in Delphi for very fast computation, see Figure 5. In the previous works [16 - 19], the code was written in MatLab. The computation speed increased 5 folds with Delphi implementation.

The iteration process terminated when the computation converges to a specified very small value, i.e. 1.0^{-10} , where there were no more significant changes in temperature values. The number of iterations, maximum error and elapsed time for various numerical techniques are shown in Table 1. It was clearly shown in Figure 6 that HSGS9L iteration performed faster than the various previous methods. Note that the speed of computation gets faster as the number of obstacles increases, since nodes occupied by obstacles were ignored during computation, see Table 2.

Once the temperature distributions were obtained, the path was generated by performing steepest descent search from the start points to the goal point. The process of generating the paths was very fast. From the current point, the algorithm simply picked the lowest temperature value from its eight neighbouring points. This process continues, until the generated path reached the goal point. As shown in Figure 7(a) - (d), all three paths are successfully generated for all four scenarios of obstacles setup.

6. Conclusions

This study shows that solving robot path planning problem using numerical techniques are indeed very attractive and feasible due to the recent advanced and

new found techniques as well as the availability of fast machine nowadays. The proposed HSGS9L iterative method performs significantly faster than the previous methods as described in our earlier works [16 - 19]. In the future work, we would include an accelerator through Successive Over-Relaxation (SOR) into the proposed method to further speed up the computation, similar to the application of Nine-Point Laplacian in [20],[21].

7. References

- [1] Connolly, C.I. & Gruppen, R. 1993. On the applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7): 931-946.
- [2] Khatib, O. 1985. Real time obstacle avoidance for manipulators and mobile robots. *IEEE Transactions on Robotics and Automation* 1:500-505.
- [3] Koditschek, D.E. 1987. Exact robot navigation by means of potential functions: Some topological considerations. *Proceedings of the IEEE International Conference on Robotics and Automation*: 1-6.
- [4] Connolly, C. I., Burns, J.B. & Weiss, R. 1990. Path planning using Laplace's equation. *Proceedings of the IEEE International Conference on Robotics and Automation*: 2102-2106.
- [5] Akishita, S., Kawamura, S. & Hayashi, K. 1990. Laplace potential for moving obstacle avoidance and approach of a mobile robot. *Japan-USA Symposium on flexible automation, A Pacific rim conf.*: 139-142.
- [6] Waydo, S. & Murray, R.M. 2003. Vehicle motion planning using stream functions. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, 2003, pp.2484-2491.
- [7] Szulczyński, P., Pazderski, D. & Kozłowski, K. 2011. Real-Time Obstacle Avoidance Using Harmonic Potential Functions. *Journal of Automation, Mobile Robotics & Intelligent Systems*. Volume 5, No 3, 2011.
- [8] Sasaki, S. 1998. A Practical Computational Technique for Mobile Robot Navigation. *Proceedings of the IEEE International Conference on Control Applications*: 1323-1327.
- [9] Daily, R. & Bevly, D.M. 2008. Harmonic Potential Field Path Planning for High Speed Vehicles. In the proceeding of American Control Conference, Seattle, June 11-13, 4609-4614.
- [10] Garrido, S., Moreno, L., Blanco, D. & Monar, F.M. 2010. Robotic Motion Using Harmonic Functions and Finite Elements. *Journal of Intelligent and Robotic Systems archive*. Volume 59, Issue 1, July 2010. Pages 57 - 73.
- [11] Abdullah, A.R. 1991. The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver. *International Journal of Computer Mathematics* 38, 61-70 (1991).
- [12] Ibrahim, A. & Abdullah, A.R. 1995. Solving the two-dimensional diffusion equation by the four point

explicit decoupled group (EDG) iterative method. International Journal Computer Mathematics, 58 (1995) 253-256.

[13] Yousif, W. S. & Evans, D.J. 2001. Explicit Decoupled Group iterative methods and their implementations. Parallel Algorithms and Applications, 7 (2001) 53-71.

[14] Abdullah, A.R. & Ali, N.H.M. 1996. A Comparative Study of Parallel Strategies for the Solution of Elliptic PDE's. Parallel Algorithms and Applications, 10 (1996) 93-103.

[15] Sulaiman, J., Hasan, M.K. & Othman, M. 2004. The Half-Sweep Iterative Alternating Decomposition Explicit (HSIADE) Method for Diffusion Equation. CIS 2004, LNCS 3314, pp. 57-63, 2004. Springer-Verlag Berlin Heidelberg.

[16] Saudi, A. & Sulaiman, J. 2009. Block Iterative Method for Robot Path Planning. The 2nd Seminar on Engineering and IT 2009 (SEIT09), Kota Kinabalu, July 7 - 8, 2009.

[17] Saudi, A. & Sulaiman, J. 2010. Numerical Technique for Robot Path Planning using Four Point-EG Iterative Method. In proc. of the 2010 Int. Symposium on Information Technology (ITSim),

June 15 - 17, 2010, Kuala Lumpur, Malaysia. Page(s): 831 - 836. ISBN: 978-1-4244-6715-0.

[18] Saudi, A. & Sulaiman, J. 2009. Half-Sweep Gauss-Seidel (HSGS) Iterative Method for Robot Path Planning. Proceedings of the 3rd International Conference on Informatics and Technology 2009 (Informatics 2009), Kuala Lumpur, Malaysia, Oct 27 - 28, 2009.

[19] Saudi, A. & Sulaiman, J. 2010. Block Iterative Method using Nine-Point Laplacian for Robot Path Planning. European Journal of Scientific Research. ISSN 1450-216X Vol.43 No.2, pp.204-211.

[20] Saudi, A. & Sulaiman, J. 2012. Robot Path Planning Using Four Point-Explicit Group Via Nine-Point Laplacian (4EG9L) Iterative Method. International Journal Procedia Engineering Volume 41, 2012, Pages 182-188. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Sep 4 - 6, 2012, Kuching, Malaysia. ISSN: 1877-7058.

[21] Adams, L.M., Leveque, R.J. & Young, D.M. 1988. Analysis of the SOR Iteration for the 9-Point Laplacian. SIAM Journal of Numerical Analysis 25 (5): 1156-1180.

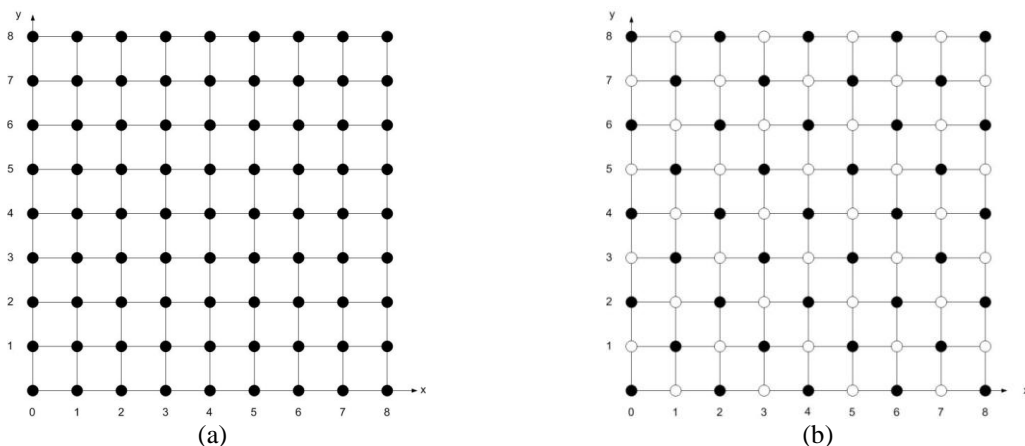


Figure 1: (a) All nodes will be considered in full-sweep iteration. (b) Only black points will be considered in half-sweep iteration.



Figure 2: Stencils of 5-point formula for (a) full-sweep and (b) half-sweep iteration, respectively.

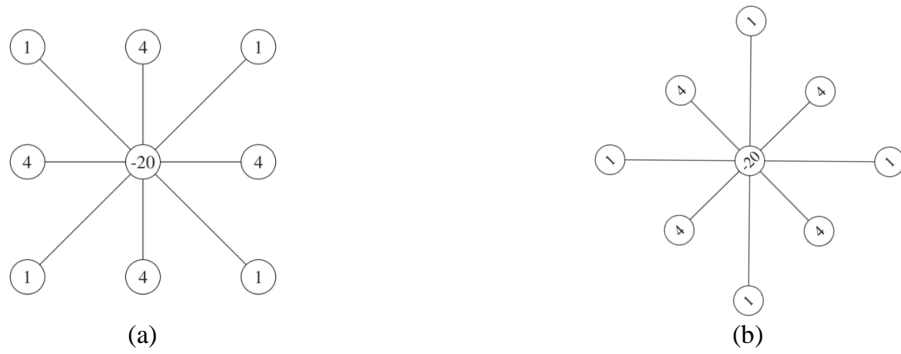


Figure 3: Stencils of 9-point formula for (a) full-sweep and (b) half-sweep iteration, respectively.

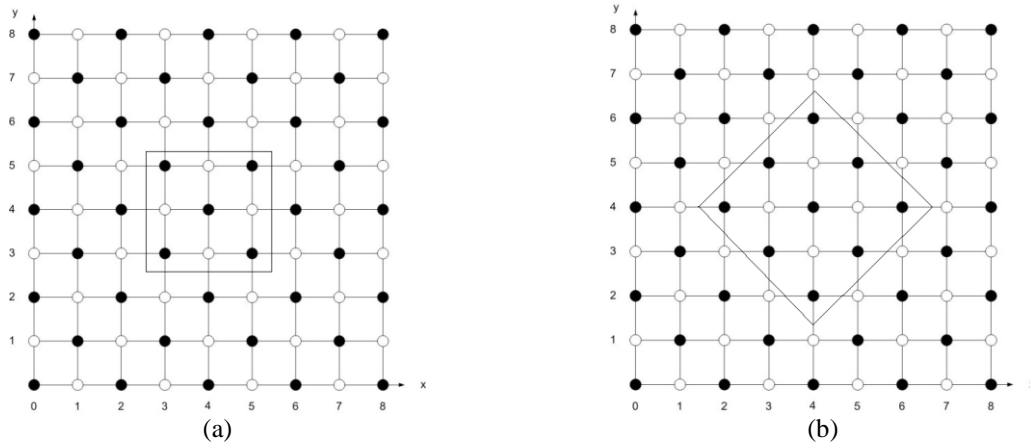


Figure 4: (a) Fives black points for each calculation of 5-point formula implementation. (b) For 9-point formula computation, nine black points are used.

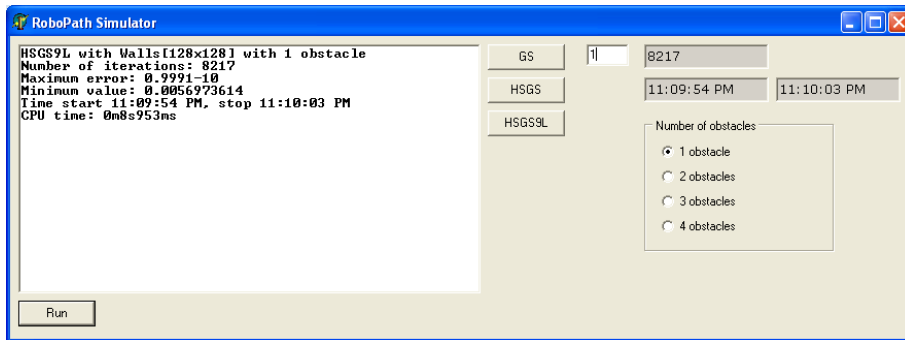


Figure 5: RoboPath Simulator was written in Delphi for very fast computation.

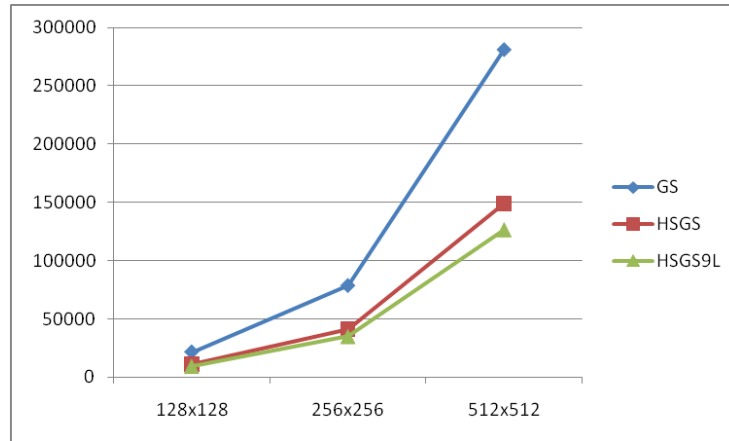


Figure 6: Graph performance of the three iterative methods. Number of iterations against sizes of environment model in 1 obstacle setup.

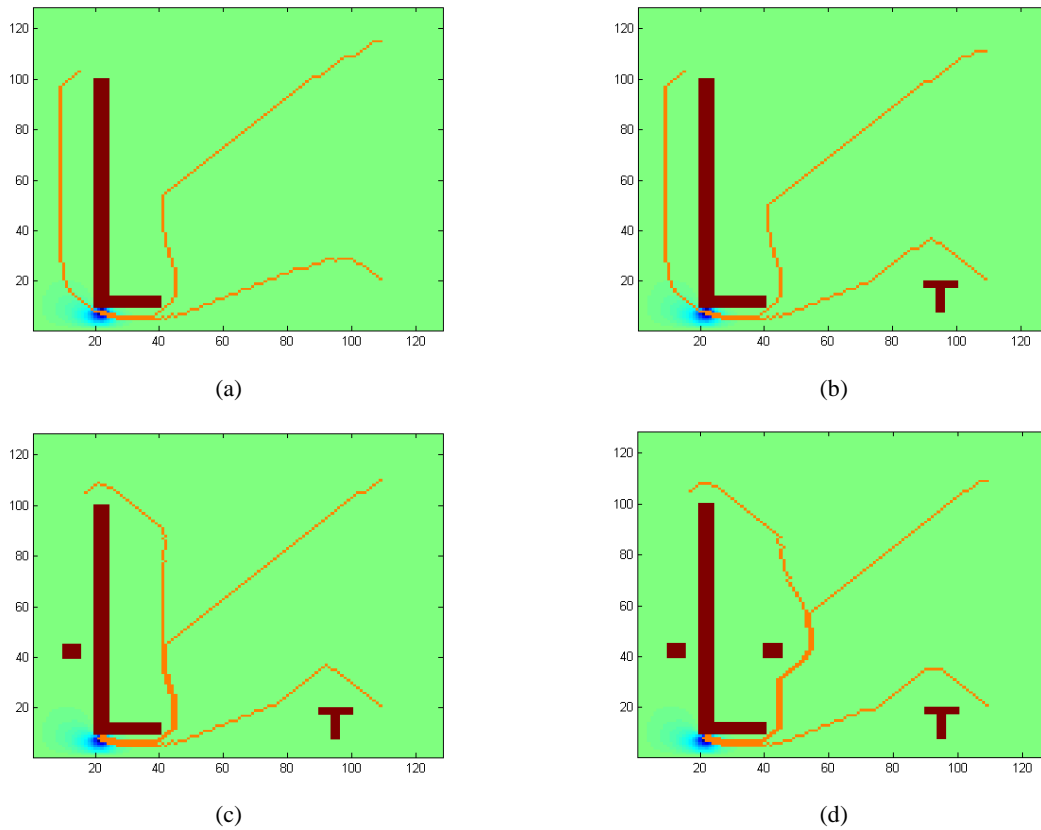


Figure 7: The paths generated with Half-Sweep Gauss-Seidel via Nine-Point Laplacian iterative method. (a) One obstacle; (b) Two obstacles; (c) Three obstacles; (d) Four obstacles.

Table 1. Performance comparison of GS vs HSGS vs HSGS9L in 1 obstacles setup.

	Iterative methods	Sizes of environment		
		128x128	256x256	512x512
Number of iteration	GS	21552	78522	281220
	HSGS	11280	41344	149130
	HSGS9L	9537	34984	126352
Maximum error	GS	0.9999 ⁻¹⁰	0.9999 ⁻¹⁰	0.9999 ⁻¹⁰
	HSGS	0.9990 ⁻¹⁰	0.9999 ⁻¹⁰	0.9999 ⁻¹⁰
	HSGS9L	0.9988 ⁻¹⁰	0.9998 ⁻¹⁰	0.9999 ⁻¹⁰
Elapsed time (m:s:ms)	GS	0:15:517	3:52:336	61:00:070
	HSGS	0:03:485	0:43:845	10:20:754
	HSGS9L	0:03:750	0:48:643	11:39:741

GS: Gauss-Seidel; HSGS: Half-Sweep Gauss-Seidel; HSGS9L: HSGS via Nine-Point Laplacian.

Table 2. Performance of HSGS9L with varying number of obstacles.

	Number of obstacles	Size of environment		
		128x128	256x256	512x512
Number of iterations	1	9537	34984	126352
	2	9268	34080	123362
	3	9073	33451	121337
	4	8132	30327	111041
Maximum error	1	0.9988 ⁻¹⁰	0.9998 ⁻¹⁰	0.9999 ⁻¹⁰
	2	0.9997 ⁻¹⁰	0.9999 ⁻¹⁰	0.9999 ⁻¹⁰
	3	0.9987 ⁻¹⁰	0.9998 ⁻¹⁰	0.9999 ⁻¹⁰
	4	0.9986 ⁻¹⁰	0.9997 ⁻¹⁰	0.9999 ⁻¹⁰
Elapsed time (m:s:ms)	1	0m3s844ms	0m48s643ms	11m39s741ms
	2	0m3s750ms	0m48s2ms	11m30s632ms
	3	0m3s672ms	0m46s907ms	11m10s506ms
	4	0m3s282ms	0m42s626ms	10m25s630ms