ARPN Journal of Science and Technology

http://www.ejournalofscience.org

Laplacian Behaviour-Based Control (LBBC) for Robot Path Planning using Explicit Group Successive Over-Relaxation via Nine-Point Laplacian (EGSOR9L) Iterative Method

[1] Azali Saudi, [2] Jumat Suleiman
[1] School of Engineering and Information Technology
[2] School of Science and Technology, University Malaysia Sabah
88400 Kota Kinabalu, Sabah, Malaysia
[1] azali60@gmail.com, [2] jumat@ums.edu.my

ABSTRACT

In this paper, we proposed a searching algorithm for generating path of a mobile robot. The method is inspired by the behaviour-based paradigm approach to robotics architecture, in which the searching algorithm employs Laplacian Behaviour-Based Control (LBBC) during space exploration of the environment. The LBBC make use of the potential function in the configuration space to guide its exploration. Laplace's equation was used to represent the potential function in the configuration space of the robot. Consequently, the gradient of the potential function would be used by the searching algorithm to generate path from starting to goal location. In complex environment, however, it suffers from the occurrence of flat region with no appreciable gradient, which result in the difficulty for the searching algorithm to generate path. The LBBC would enable the searching algorithm to generate path successfully even with the occurrence of this flat region. In this paper, the solution to Laplace's equation is calculated with block iteration via Explicit Group Successive Over-Relaxation via Nine-Point Laplacian (EGSOR9L) iterative method for rapid computation compared to the traditional Gauss-Seidel iteration and standard SOR.

Keywords: *Laplacian Behaviour-Based Control (LBBC), Robot path planning, Explicit Group Successive Over-Relaxation via Nine-Point Laplacian (EGSOR9L), Laplace's equation.*

1. INTRODUCTION

Robust autonomous path planning capability is essentially one of the most useful features for any robotics application. A truly autonomous mobile robot must have the capability to efficiently and reliably plan a route from start to the goal point without colliding with obstacles in between.

Path planning algorithm attempts to deal with the problem of establishing a medium of communication between initial and final configurations, so that the robot can traverse the field safely. Various algorithms exist trying to solve this problem but all have shortcomings. The difficulty is due to the complexity of path planning problem, where it increases exponentially with the dimension of the configuration space.

In order to ensure completeness, every point in the configuration space has to be considered in the computation. Many global path planning methods presuppose a complete representation of the configuration space. Their main drawbacks, is that at best they are computationally expensive and often intractable. Potential field and bug approaches are local methods that do not make this assumption but are not complete methods. Thus, produce the occurrence of local minima or loops that will often cause this class of path planners to fail.

This work attempts to solve robot path planning problem by employing global method to generate path in complex environment. By applying Laplacian Behaviour-Based Control (LBBC), a robust searching algorithm will quickly generate path from starting to goal point configuration. Based on the theory of heat transfer, the environment is modelled as a configuration space, in which temperature distribution at each point will be used by the LBBC to guide its searching. The solutions of Laplace's equation, also known as harmonic functions, can be used to represent temperature values in the configuration space. For fast computation, the temperature distribution is numerically computed by solving Laplace's equation using weighted iterative method. In this work, several experiments were conducted to study the performance of using weighted iterative method for computing the solution of Laplace's equation, and to investigate the effectiveness of LBBC to generate path in several sizes of environment with varying setup of obstacles.

2. LITERATURE REVIEW

Connolly and Gruppen [1] reported that harmonic functions have a number of properties useful in robotic applications. The use of potential functions for robot path planning, as introduced by Khatib [2], views every obstacle to be exerting a repelling force on an end effectors, while the goal exerts an attractive force. Koditschek [3], using geometrical arguments, showed that, at least in certain types of domains, there exists potential functions which can guide the effectors from almost any point to a given point. These potential fields

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

approach to path planning, however, suffer from the spontaneous creation of local minima.

Connolly et al. [4] and Akishita et al. [5] independently developed a global method using solutions to Laplace's equations for path planning to generate a smooth, collision-free path. The potential field is computed in a global manner, i.e. over the entire region, and the harmonic solutions to Laplace's equation are used to find the path lines for a robot to move from the start point to the goal point. This global method, however, suffer from the occurrence of flat region in complex environment which caused the path generation algorithm to fail.

Several other methods are also proposed for solving path planning problem. In [6], an algorithm that employs distance transform method is reported. Jan et al. [7] conducted researches on utilizing geometry maze routing algorithm. The work by Bhattacharya and Gavrilova [8] uses Voronoi Diagram to solve path planning problem. In [9], genetic algorithm through evolutionary process was used for mobile robot path planning.

3. LAPLACIAN BEHAVIOUR-BASED CONTROL

Traditional approach robot programming assumes the availability of a complete and accurate model of the robot and its environment, relying on planners to generate actions [10]. Unfortunately, this approach has several disadvantages. One main drawback is that they require huge amounts of computational resources. This drawback is much obvious for an autonomous mobile robot that must carry its own computational resources. Secondly, this approach must be based on highly accurate model, thus it requires a number of high-precision sensors which are also often expensive. These sensors, however, are subject to noisy data. Finally, this sense-plan-act paradigm is by nature sequential, thus it would fail if the world happens to change in between of phases. Furthermore, there is always delay between sensing and act, due to longer time required in planning.

As an alternative to the traditional approach, a new paradigm called subsumption architecture, also known as behaviour-based control, is devised [11]. In this architecture, sensors are dealt with only implicitly in that they initiate behaviours. Each behaviour is simply layers of control systems that all run in parallel. Higher level behaviours have the power to temporarily suppress lower level behaviours. Therefore, a set of priority scheme is used to resolve the dominant behaviour for a given scenario. A more rigorous explanation of behaviour-based approach for controlling robot is presented in [12].

In this work, inspired by the behaviour-based paradigm approach to robotics control, the searching algorithm employs Laplacian Behaviour-Based Control (LBBC) for robust space exploration of the configuration space. The LBBC comprises four core behaviours i.e. keep-forward, follow-wall, avoid-obstacle, and find-slope. All these core behaviours make use of the potential values represented by temperature distribution in the configuration space which are computed numerically to provide guidance during search exploration.

a. Keep-Forward Behaviour

The keep-forward behaviour is a core behaviour that keeps the searching moving forward in the same direction as long as the temperature at current location is higher than the next location. When the searching encounters ascending slope, flat region, obstacles or walls, the keep-forward behaviour stops, and other behaviours would take over. The main aim of this behaviour is to guide the searching by following the descending slope until the goal location is found.

b. Follow-Wall Behaviour

The follow-wall behaviour provides the search with the capability to follow the wall for a specified number of steps. With this behaviour, it will command the searching to keep turning gradually until its direction is parallel with the wall. It provides the searching with the capability of traversing the narrow path and sharp corner. In this implementation, the follow-wall behaviour is executed for every a specified number of steps. After that the searching switches to find-slope behaviour.

c. Avoid-Obstacle Behaviour

When the searching hits an obstacle or wall, it will trigger the searching to backup and turn 90 degrees to the left or right alternately. By turning alternately to the left and right, it provides the searching with the capability to escape from a difficult position such as sharp corner.

d. Find-Slope Behaviour

When the find-slope behaviour takes over, it will command the searching to move randomly hoping to encounter a descending slope that consequently triggers keep-forward behaviour. With this behaviour, the searching is capable of moving away from a flat region to continue its descending move towards goal location.

4. HARMONIC FUNCTIONS

A harmonic function on a domain is a function which satisfies Laplace's equation,

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

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

where x_i is the i-th Cartesian coordinate and n is the dimension. In the case of robot path construction, the boundary of Ω (denoted by $\partial\Omega$) consists of the outer boundary of the workspace and the boundaries of all the obstacles as well as the start point and the goal point, in a configuration space representation. The spontaneous creation of a false local minimum inside the region Ω is avoided if Laplace's equation is imposed as a constraint on the functions used, as the harmonic functions satisfy the min-max principle. Laplace's equation can be solved numerically. Standard methods are Jacobi and Gauss-Seidel, whilst in this paper Eq. (1) was solved via block iterative method using 4 Point-EG for faster computation.

5. Configuration Space

In the framework used in this study, the robot is represented by a point in the configuration space, or C-space. The path planning problem is then posed as an obstacle avoidance problem for the point robot from the start point to the goal point in the C- space. The C-space can have either square or rectangular outer boundaries, having projections or convolutions inside to act as barriers. Apart from projections of the boundaries, some obstacles inside the boundary are also considered. The C-space is designed in grid or discrete form and the coordinates and function values associated with each node are computed iteratively by applying 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. In some cases with Dirichlet conditions, the start point is not assigned any temperature. In this study, Dirichlet boundary conditions are employed, thus the results are processed by assigning different temperature values to the boundaries and obstacles, and lowest temperature for the goal point. No temperature values are assigned to the start points. In this work, solution to the Laplace's equation were subjected to Dirichlet boundary conditions, $\Phi \mid \partial\Omega = c$, where c is constant.

6. The Formulation of Explicit Group Successive Over-Relaxation via Nine-Point Laplacian (EGsor9L) Iterative Method

In the literature, Jacobi, Gauss-Seidel and SOR [13] had been used for solving any linear system. More recently, Daily and Bevly [14] use analytical solution for arbitrarily shaped obstacles. In this study, we employ block iterative method via Explicit Group with SOR on Nine-Point Laplacian (also known as EGSOR9L) for solving the Laplace's equation. Actually, many discussions of the block iterative methods mainly on various points of Explicit Group have been explained by Evans [15], Evans & Yusuf [16], Ibrahim [17]. Furthermore, a modified Explicit Group method via EDGSOR was used by Suleiman et al. [18]. They pointed out that block iterative methods perform much faster than the standard point iteration.

Let us consider the two-dimensional Laplace equation in Eq. (1) defined as

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

The discretization of Eq. (2) based on 9-point formula (now known as Nine-Point Laplacian) can be shown as below

$$\begin{aligned} &4(U_W + U_E + U_N + U_S) + D_{BL} + D_{BR} + D_{TL} + D_{TR} - 20U_C = 0, \\ &W = i-1,\ j;\ E = i+1;\ N = i,\ j+1;\ S = i,\ j-1; \\ &BL = i-1,\ j-1;\ BR = i+1,\ j-1; \\ &TL = i-1,\ j+1; TR = i+1,\ j+1; \\ &C = i,\ j \end{aligned} \qquad (3)$$

The equation in Eq. (3) shown above is the standard 9-point formula of Gauss-Seidel iterative method for solving linear system.

Then, with this 9-point formula, let us consider a block of two node points as shown in Figure 1 and defined as

$$\begin{bmatrix} 20 & -4 \\ -4 & 20 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \qquad (4)$$

where

$$S_1 = 4(U_W + U_S + U_N) + D_{BL} + D_{BR} + D_{TL} + D_{TR},$$
$$S_2 = 4(U_{i+2,j} + D_{BR} + D_{TR}) + U_S + U_{i+2,j-1} + U_N + U_{i+2,j+1}.$$

Determining the inverse matrix of the coefficient matrix in Eq. (4), the general scheme of Eq. (4) can be rewritten as:

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \end{bmatrix} = \frac{1}{96} \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \qquad (5)$$

The implementation of Eq. (5) can be shown as

$$U_{i,j} = \frac{1}{96}(5S_1 + S_2),$$
$$U_{i+1,j} = \frac{1}{96}(S_1 + 5S_2). \qquad (6)$$

By adding a weighted parameter ω to Eq. (6), the implementation of this iterative method can be shown as (Young [19]):

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

$$U_{i,j}^{(k+1)} = \frac{\omega}{96}(5S_1 + S_2) + (1-\omega)U_{i,j}^{(k)} ,$$
$$U_{i+1,j}^{(k+1)} = \frac{\omega}{96}(S_1 + 5S_2) + (1-\omega)U_{i+1,j}^{(k)} . \quad (7)$$

7. Experiments And Results

The experiment considered various size of static environment, i.e. 128x128, 256x256, and 512x512, that consists of a goal point, several starting points and varying setup of walls and obstacles. Initially, the boundaries (walls) and obstacles were fixed with high temperature values. Goal point was set to very low temperature. All other free spaces were set to zero temperature value. Then, the iteration process was run on Intel Core 2 Duo CPU running at 1.83GHz speed with 1GB of RAM to compute temperature values numerically at all points in the environment. The iteration process was terminated when there was no more changes in temperature values, where it converged to a specified very small value, i.e. 1.0^{-10}. The highest precision of solution for Eq. (1) was required to reduce the occurrence of flat area, hence would speed up the searching algorithm during path planning construction of the mobile robot from starting point to goal point. Table I shows the number of iterations, maximum error and elapsed time (in m:s:ms) for computing all temperature values in the environment. Clearly, EGSOR9L iteration proved to be very fast compared to the previous iterative methods.

Once the temperature values were obtained, the searching algorithm would make use of them to guide its exploration. In the previous works by Saudi [20,21,22,23], the path can be generated successfully even without LBBC, if the environment space was simple and sparse in which the gradient from start points to goal point are smooth, see Figure 3(a). However, the searching algorithm failed to reach the goal point when the horizontal wall was extended. As shown in Figure 3(b), only one path was successfully generated, whereas the other two start points got stuck in the flat region. By employing LBBC as reported by Saudi [24], the searching algorithm would be able to escape from flat region and continue its exploration by utilizing find-slope behaviour until it detects a wall in which the algorithm switches to follow-wall behaviour, see Figure 3(c). The LBBC algorithms then switches back to keep-forward behaviour until it reaches the goal point, see Figure 3(d).

8. Conclusions

The experiment in this study shows the effectiveness of the Laplacian Behaviour-Based Control (LBBC), which was first introduced by Saudi in [24] to generate path for mobile robot in varying setup and sizes of environment model. Unlike previous methods in [20,21,22,23], the LBBC [24] provides the searching algorithm with the capability to escape from flat region and difficult position, thus the searching algorithm could continue its move towards goal location. Moreover, the potential values of each point in the configuration, i.e. the solution of Laplace's equation, as shown in Eq. (1), can be computed rapidly by using block iterative method which is faster than the previous standard point Gauss-Seidel iteration. The computation speed up can be improved dramatically by imposing SOR to the formula. By computing with Nine-Point Laplacian, the number of iteration was further reduced by more than 20% (EGSOR vs EGSOR9L) , see Table I. Future work would investigate the performance of utilizing Explicit Group on four points and half-sweep iteration for solving path planning problem in robotics application.

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 conference: 139–142.

[6] Willms, A.R. and Simon X.Y. 2008. IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 38. No. 3, June 2008. Real-Time Robot Path Planning via a Distance-Propagating Dynamic System with Obstacle Clearance.

[7] Jan. G.E., Chang, K.Y., and Parberry I. 2008. Optimal Path Planning for Mobile Robot Navigation. IEEE/ASME Trans. on Mechatronics, Vol. 13, No. 4, Aug 2008, pages 451-460.

[8] Bhattacharya, P. and Gavrilova, M.L. 2008. Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path. IEEE Robotics & Automation Magazine, Volume 15, Issue 2, June 2008. Page(s):58 – 66.

ARPN Journal of Science and Technology

[9] Chen, W. And Qin, H. 2011. Path planning of mobile robot based on Hybrid Cascaded Genetic Algorithm. In proc. of the 9th World Congress on Intelligent Control and Automation (WCICA), 21-25 June 2011. Page(s): 501 – 504. ISBN: 978-1-61284-698-9

[10] Brooks, R. A. 1986. A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, 2 :(14-23).

[11] Arkin, R. C. 1998. Behaviour-based robotics, Bradford Books.

[12] Saudi, A. & Hallam, B. 2004. Tale of Two Behaviour-based Robots. Mindstorms vs Technic. In proc. of IEEE Region 10 Conference (TENCON 2004), 21-24 Nov. 2004. Page(s): 479 - 482 Vol. 4. ISBN: 0-7803-8560-8.

[13] Sasaki, S. 1998. A Practical Computational Technique for Mobile Robot Navigation. Proceedings of the IEEE International Conference on Control Applications: 1323-1327.

[14] 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.

[15] Evans, D. J. 1985. Group Explicit Iterative methods for solving large linear systems. Int. J. Computer Maths., 17: 81-108.

[16] Evans, D.J & Yusuf, W. S. 1986. Explicit Group Iterative Methods for solving elliptic partial differential equations in 3-space dimensions. Int. J. Computer Maths., 18:323-340.

[17] Ibrahim, A. 1993. The Study of the Iterative Solution of Boundary Value Problem by the Finite Difference Methods. PhD Thesis. University Kebangsaan Malaysia.

[18] Sulaiman, J., Hasan, M.K. & Othman, M. 2007. Red-Black EDGSOR Iterative Method Using

[19] Triangle Element Approximation for 2D Poisson Equations. In. O. Gervasi & M. Gavrilova (Eds). Computational Science and Its Application 2007. Lecture Notes in Computer Science (LNCS 4707): 298-308. Berlin: Springer-Verlag.

[20] Young, D.M. 1971. Iterative solution of large linear systems. London: Academic Press.

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

[22] Saudi, A. & Suleiman, J. 2009. Efficient Weighted Block Iterative Method for Robot Path Planning Using Harmonic Functions. Proceedings of the 2nd International Conference on Control, Instrumentation and Mechatronic Engineering (CIM09), Malacca, Malaysia, June 2-3, 2009.

[23] Saudi, A. & Suleiman, 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 (2010), pp.204-211.

[24] Saudi, A. & Suleiman, J. 2012. Robot Path Planning via EGSOR Iterative Method using Nine-Point Laplacian. In proc. of the 3rd International Conference on Intelligent Systems, Modelling and Simulation (ISMS 2012), Kota Kinabalu, Malaysia. Issue Date: Feb 8-10, 2012, page(s): 61 – 66. ISBN: 978-1-4673-0886-1.

[25] Saudi, A. & Suleiman, J. 2012. Laplacian Behaviour-Based Control for Robot Path Planning using Full-Sweep Successive Over-Relaxation via Nine-Point Laplacian (FSSOR9L). International Journal of Applied Science and Technology, Vol. 2 No. 3; March 2012, pp: 255 - 261. ISSN 2221-0997 (Print), ISSN 2221-1004 (Online).

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

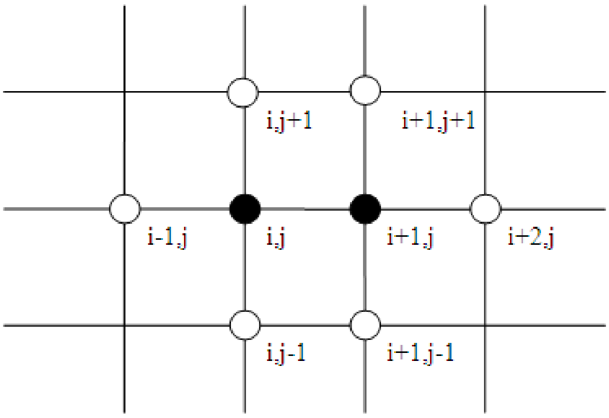


Fig 1: Illustration of a block of two node points to be computed simultaneously.

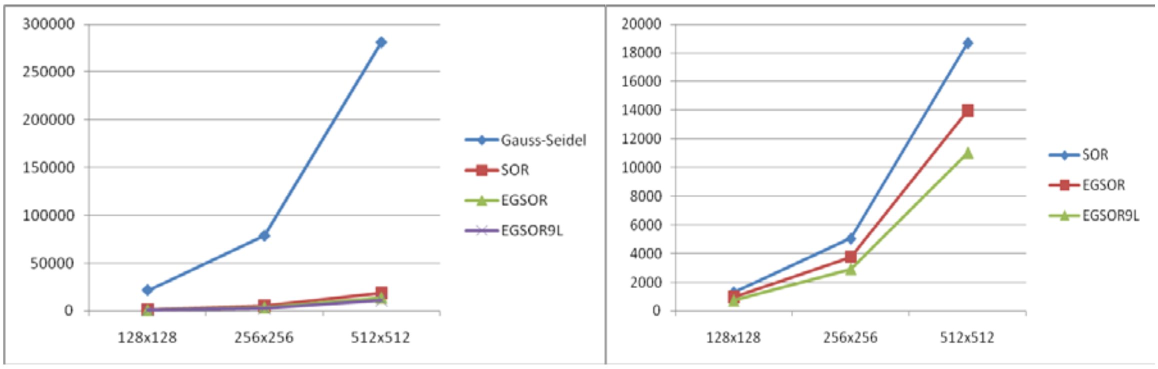
TABLE 1: PERFORMANCE COMPARISON OF SEVERAL ITERATIVE METHODS AGAINST VARYING SIZE OF ENVIRONMENT.

		Size of environment		
		128x128	256x256	512x512
Number of iterations	GS	21552	78522	281220
	SOR	1314	5059	18699
	EGSOR	962	3768	13982
	EGSOR9L	723	2924	11034
Maximum error	GS	0.9993^{-10}	0.9988^{-10}	0.9996^{-10}
	SOR	0.9952^{-10}	0.9985^{-10}	0.9998^{-10}
	EGSOR	0.9897^{-10}	0.9983^{-10}	0.9998^{-10}
	EGSOR9L	0.9842^{-10}	0.9995^{-10}	0.9998^{-10}
Elapsed time (m:s:ms)	GS	0:21:297	6:28:235	49:18:32
	SOR	0:1:344	0:24:641	7:11:188
	EGSOR	0:1:109	0:19:828	5:39:250
	EGSOR9L	0:0:594	0:10:797	2:57:531

GS: Gauss-Seidel; SOR: Successive Over-Relaxation;
EGSOR: Explicit Group with SOR; EGSOR9L: EGSOR on Nine-Point Laplacian.

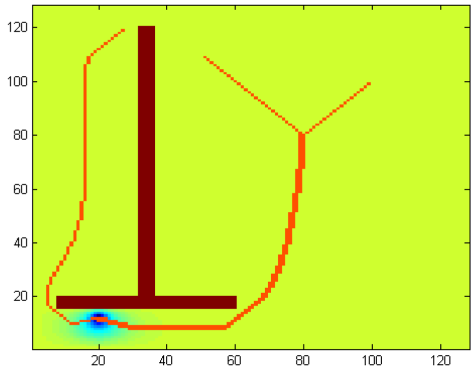
ARPN Journal of Science and Technology

http://www.ejournalofscience.org

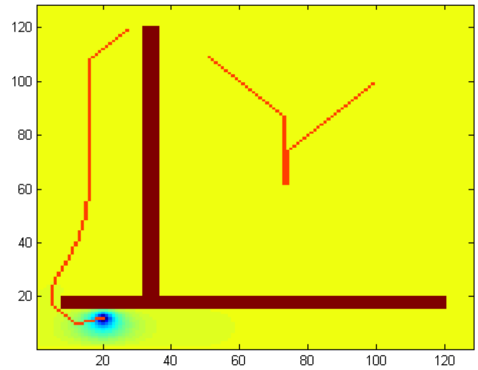


(a) (b)

Fig 2: (a) Graph of the number of iteration against size of environment for various iterative methods.
(b) The same graph of (a), but without Gauss-Seidel for better visual comparison of SOR vs EGSOR vs EGSOR9L.

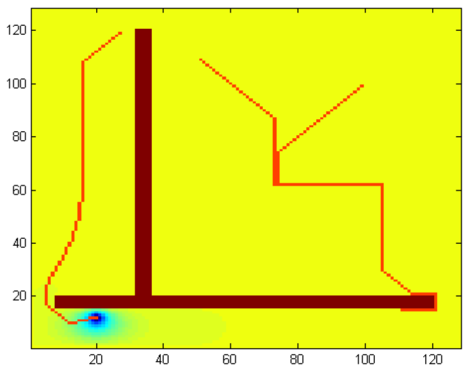


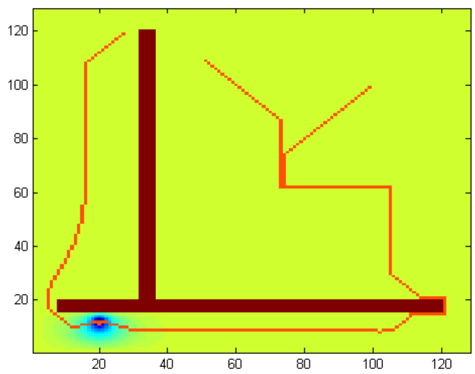
(a)



(b)

(b)





(c) (d)

Fig 3: (a) Path is successfully generated in a simple and sparse environment, from three starting points to a goal point.
(b) The path generation process failed to reach the goal point when the length of horizontal wall is extended twice to the right.
(c) With LBBC, the algorithm switches to find-slope behaviour and then follow-wall behaviour to escape from flat region.
(d) The LBBC algorithm switches back to keep-forward behaviour to find the goal point.