Finite Difference Vehicular Path Planning

John Wharington and Israel Herszberg

(*Royal Melbourne Institute of Technology*)

This paper presents a novel approach to computing minimum-time paths based on a technique originally developed for use in geophysics. The technique is based on a finite difference scheme and is quite efficient in terms of both computing effort and storage. A particular strength of the technique is that it provides optimal paths to all locations in the field, thus being useful in situations where the goal is not known at the outset of the search. Details are presented on the basic technique and extensions are derived to include the integrations of resources (such as fuel store), and to provide flexibility in the objective. Two examples are given: minimum-time planning for vehicles under the influence of drift, and planning to minimise the risk of detection for a stealthy vehicle in the presence of threats.

1. INTRODUCTION. The determination of the optimal course to a destination is one of the major goals of navigation. This problem has received considerable attention, and now high-speed computers can be used to find paths that, for example, minimise the distance or time of travel. Path planning is of obvious importance with regard to the efficient use of commercial and military vehicles.

Navigators and route planners are by no means alone in having an interest in the path planning problem. Algorithms for determining the shortest path in a network of nodes are employed in such diverse applications as critical path analysis, factory scheduling, automatic speech recognition and control of robotic arms,[1] to name but a few. Seismologists have a vested interest in highly efficient techniques for determining minimum time paths, which facilitate inverse tomography – the determination of geological features from sounding data.

In this paper, we adapt a Finite Difference (FD) approximation to ray tracing, initially developed in geophysics, for the purposes of vehicular path planning. The most basic form of the technique computes minimum time paths where the vehicle speed can vary with direction, location and time; we also show how the technique can be extended to compute paths that satisfy other objectives.

This paper is organised as follows. First, we present the finite difference path-planning method and discuss its implementation on a single processor and parallel computers. Next, extensions to the method are presented to integrate vehicle resources during planning, and to transform the minimum-time objective. Demonstrations are given of minimum-time planning for a vehicle influenced by drift, and for minimum-risk planning with respect to a stealthy vehicle. Finally, we discuss the merits of this approach in terms of efficiency and candidate applications.

2. FINITE DIFFERENCE PATH PLANNING. By far the most widely used path-planning techniques are graph-theoretic, in which the problem is formulated as a search of arcs that minimise the distance between an origin and destination, in a discrete workspace. Searching the domain of possible solutions is

reduced with heuristics, such as the Dijkstra and auction algorithms.[2,3] This approach is quite general to the objective; for instance, minimum-time paths can be obtained by replacing the arc length by traversal time.

Alternatively, the problem of computing minimum-time paths for vehicles over large distances can be regarded as the same problem as the simulation of wave propagation. According to Fermat, waves refract in heterogeneous media so as to follow the path of least resistance. The analogy is made workable by modelling the dependence of vehicle speed to direction and the type of media. As we shall see in Section 4, working with minimum-time schemes need not exclude other objectives.

Vidale pioneered the finite difference approach to simulating wave propagation.[4] His method is based on computing a field of arrival times, the total time taken to reach free locations on the field. Working with an arrival field is convenient as it allows temporal dependencies of the environment or vehicle to be taken into account. Secondly, the vehicle resources can be readily integrated over this field, producing a map of resources for use by a system for planning goals, and where the vehicle speed depends on resource levels.

A brief outline of the method is given here for the case of planning in two dimensions on a regular grid; for further details the reader is referred to Reference 4 for derivation of the isotropic case (speed varies with location only) and Reference 5 for the extension to anisotropy (speed varies with location and direction). The method can be extended to three or more dimensions and for use with irregular grids.

2.1. *Node Calculation.* The finite difference method is an approximation to the eikonal equation of ray tracing, shown here for a two-dimensional workspace of coordinates x and y:

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 = q(x, y)^2, \tag{1}$$

where q is the slowness (inverse velocity) of rays and t arrival time.

Anisotropy is represented by a slowness model $f(q_x, q_y) = 0$ where q_x and q_y are slowness in the x and y directions. Associated with this model is the ray angle ψ which is normal to the slowness surface. Each cell is assumed locally homogeneous, but otherwise heterogeneity is represented by slowness surfaces that vary with position.

There are two modes of ray propagation: head waves, which issue from a single node; and plane waves, which depend on the relative arrival time of a neighbouring pair of nodes. The minimum arrival time property is achieved implicitly, through Fermat's principle: the path of a ray (vehicle) is such that it renders the time between source and receiver a minimum. During node updates, arrival times are calculated with the finite difference approximation from each neighbour and for each propagation mode; the lowest of these is then selected.

For head waves, calculation of the traversal time is straightforward. The slowness model is consulted to determine the minimum magnitude of slowness in the propagation direction, $|q(\psi)|$. The arrival time to the target node j from node i is:

$$t_j = t_i + h_{i,j}|q(\psi)|, \tag{2}$$

where $h_{i,j}$ is the distance from node i to j. An example is illustrated in Figure 1(a).

Determining arrival times for interference waves is slightly more involved. A template is presented for the case illustrated in Figure 1(b) and where the grid is regular. The slowness in the x direction is known from the relative arrival times of

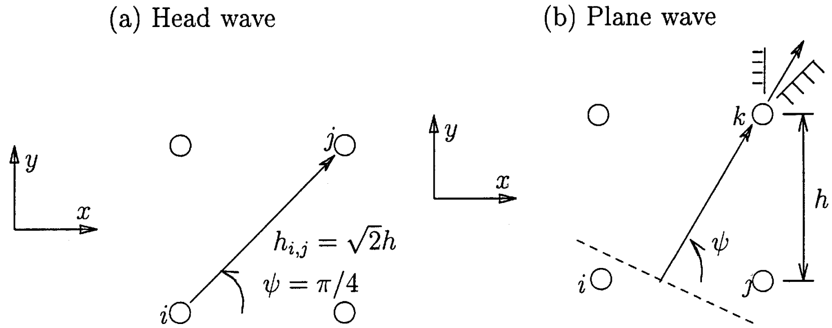

Figure 1. Wave Propagation Stencils.

nodes i and j, and we wish to determine the traversal time in the perpendicular direction, towards the target node k. The slowness model is consulted to determine the minimum slowness in the y direction and the ray angle, that correspond to the specified slowness in the x direction. Then the arrival time at the target node is:

$$t_k = t_j + hq_y. \tag{3}$$

This mode is valid if the ray reaches k from a virtual wave source along the line from i to j, that is, for nodes on a square grid, if

$$\frac{\pi}{4} \leqslant \psi \leqslant \frac{\pi}{2}. \tag{4}$$

Once a ray propagation is admitted, the associated ray direction must be recorded, since the optimal trajectory is determined by back-tracing ψ.

Hard obstacles are dealt with by assigning an infinite slowness to nodes inside obstacles. Nodes near the perimeters may be assigned high slowness to discourage paths from grazing the obstacle.

2.2. *Field Sweep*. The order in which nodes are updated depends on the computing architecture, though in general the order is chosen to be consistent with causality. That is, updating should tend to evolve in a similar pattern as the waves that are generated. Vehicle resources, such as fuel stores, can be evaluated in parallel with the field sweep. This is necessary when the vehicle speed depends on these resources.

A method for single processor computers was developed by Vidale and subsequently refined by Lecomte.[5] This latter method is the Expanding Contracting Ring Process (ECRP), so-called as it consists of a series of updates on a ring that propagates away from the origin and returns to correct for reversal of flow.

ECRP is quite simple, yet effective, even in highly heterogeneous and anisotropic media. The sequence of wave calculations are illustrated in Figure 2. Propagation consists of performing a series of updates along each edge of the ring. The first sweep (Stage I) checks head waves in the direction of ring expansion, on all nodes in any order. Stages II and III comprise diagonal head wave and plane wave checks, performed along the ring in both directions.

Once a ring has been updated, the sweep proceeds outwards, unless a head wave was generated along the ring (i.e. a head wave was admitted in Stage III). This signals diffraction, which triggers a sweep backwards (ring contraction) to accommodate

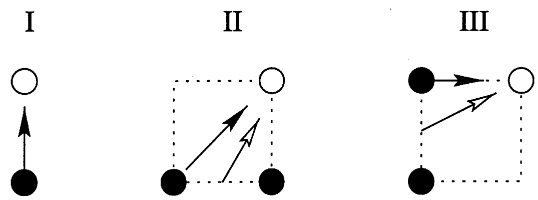

Figure 2. Global Finite Difference update scheme (after Lecomte 1993). Cells with arrival times calculated from the previous ring sweep are indicated by closed circles. Head waves are marked by closed arrow heads; open arrows indicate plane waves. The sweep direction is towards the top of the page.

reversal of rays with respect to the expansion direction. When the contracting ring encounters no revised arrival times, the ring expands outwards again.

The stability of this mechanism is yet to be formally assessed. The diffraction contingency may result in a feedback loop that can cause significant delays or even fail to converge. Nevertheless, diffraction events, while not especially rare, are generally bound in practice to the neighbourhood of their source.

An asynchronous parallel scheme has been implemented by Podvin and Lecomte,[6] whereby nodes are updated in a completely random order. This method is wasteful, since nodes then have to be updated many times before reaching a steady state. Higher speeds could be obtained with partial synchronisation, integrating a causality heuristic. For example, a processor could be assigned to each node in an ECRP ring.

For planning applications, various heuristics might be applied to process nodes according to salience criteria. A version of 'best-first' search could be employed by assigning processors to nodes drawn from a queue. When a node's arrival time changes, its neighbours post a request for processing. The priority of each node's request is assigned according to the maximum of its neighbour's arrival times. However, such a scheme would result in an incomplete planner and should be used with care. On the other hand, a 'worst-first' queue would satisfy completeness and might be superior to parallel ECRP.

2.3. *Slowness Model Queries*. The most expensive part of the method for computation is the calculation of slowness model queries. Queries relating to plane wave propagation are root-finding problems, and a one-dimensional search may be required for head-wave queries. Moreover, for general anisotropy, both of these problems cannot be solved locally, since multiple solutions may exist, of which the fastest must be chosen.

Efficient implementations therefore require solutions to these problems that are either analytic or tailored to the anisotropy particular to the application. The circular slowness model is trivial to solve. Analytic solution for elliptical slowness can be easily obtained since the plane wave problem amounts to solving a cubic.

In more challenging cases, various strategies might be employed for efficiency. The choice of which strategy to use will be motivated by the slowness model geometry and accuracy requirements. For aerospace applications, a suitable and efficient strategy is to represent the model by a piecewise elliptic function. Another strategy is to compile solutions off-line and use a corresponding function approximator on-line.

3. INTEGRATION OF RESOURCES.

A scheme is required to approximate the integration of resources along trajectories, by an integration across nodes in the planning field. The rate of resource expenditure, $\dot{E}$ is assumed to be constant

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

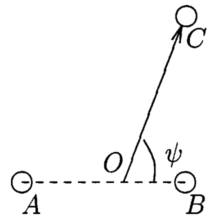

Figure 3. Calculation of Resources and Actual Arrival Times for Plane Waves.

between grid points. The integration constitutes an additional FD process that is concurrent with calculation of actual arrival times.

This is straightforward for head waves. By first order integration, the resource at the destination B is given by

$$E_B = E_A + \dot{E}\Delta t, \tag{5}$$

where $\Delta t = lq$, l being the path length.

For plane waves the scheme is more involved (see Figure 3). The resource at C is calculated by integration along OC:

$$\left.\begin{aligned} E_C &= E_O + \dot{E}\Delta t, \\ \Delta t &= hq/\sin(\psi), \end{aligned}\right\} \tag{6}$$

where h is the grid spacing. The resource at O is found by linear interpolation between the values at A and B, thus:

$$\begin{aligned} E_O &= E_A + \rho(E_B - E_A), \\ \rho &= \frac{|AO|}{|AB|}, \\ &= 1 - |\cot(\psi)|. \end{aligned} \tag{7}$$

4. TRANSFORMATION OF OBJECTIVES. The Finite Difference Path Planning (FDPP) scheme can be extended so that the minimum time property is transformed to incorporate or include secondary objectives. It may be required, for example, to avoid certain regions that may be dangerous or costly in vehicle resources.

We introduce the notion of virtual slowness, whereby an operator modifies the slowness model according to the secondary objectives. This exploits the attractive (repulsive) effect of refraction in paths passing through progressively faster (slower) regions. In the simplest case, the slowness operator is a scaling factor. Other operators that polarise the planning wave can be developed for directionally sensitive objectives. The general idea is that the objective is reflected in the choice of wave model, then arrival times are in some sense virtual, but the ray directions are correct in satisfying the objective. Additional calculation is required to keep track of actual arrival times.

There are now two arrival time fields. The first contains virtual arrival time, in which computations are based on virtual slowness. The second field contains actual arrival times of rays that propagate in the direction calculated in the first field. Obviously, the actual arrival times are used in determining the environment state during the field sweep.

For head waves, the actual arrival time between two nodes A and B is simply

$$t_B = t_A + lq_a, \tag{8}$$

where q_a is the actual slowness and l the path length.

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

For plane waves (see Figure 3), the arrival time at C is calculated by integration along OC, exactly the same way as in the integration of resources:

$$\left.\begin{aligned} t_C &= t_O + \Delta t, \\ \Delta t &= hq_a/\sin(\psi), \end{aligned}\right\} \tag{9}$$

where h is the grid spacing. The time at O is found by linear interpolation between the values at A and B, thus

$$\begin{aligned} t_O &= t_A + \rho(t_B - t_A), \\ \rho &= \frac{|AO|}{|AB|} \\ &= 1 - |\cot(\psi)|. \end{aligned} \tag{10}$$

5. DEMONSTRATION. In this section the basic and extended FDPP is demonstrated on two tasks, respectively minimum time paths of a vehicle influenced by drift, and minimum risk paths on the basis of stealth.

5.1. *Drift*. An elliptical approximation to the slowness surface that corresponds to the drift model can be obtained by solving for zero error at the intercepts at major and minor axes. Where the vehicle speed relative to the medium is V and the speed of the medium is w, this yields:

$$\left(\frac{V^2 - w^2}{V} q_x + \frac{w}{V}\right)^2 + (Vq_y)^2 = 1. \tag{11}$$

The distortion associated with this model is within engineering accuracy for $w/V < 0{\cdot}5$. For example, the speed polar for $w/V = 0{\cdot}4$ is shown in Figure 4. Note that this

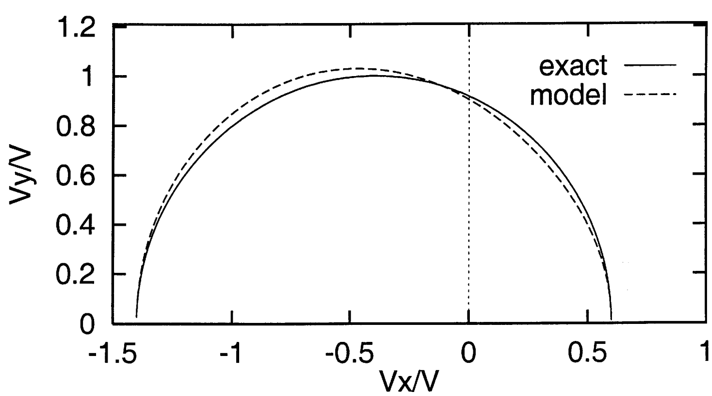

Figure 4. Elliptical Drift Model fit $w/V = 0{\cdot}4$.

model is not the best possible fit, but the expression above simplifies matters for this demonstration.

A potential sink is located at the centre of the field, for which the wind speed at the point $\vec{X} = \{x, y\}$ is

$$|w| = \frac{600}{|\vec{X} - \vec{S}| + 30}, \tag{12}$$

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

where $\vec{S}$ is the location of the sink. The wind acts in the direction $\vec{X}-\vec{S}$. The resulting arrival time contours (isochrones) and streamlines (trajectories) are illustrated in Figure 5. The vehicle speed is 25 units and the field is 600 units square.

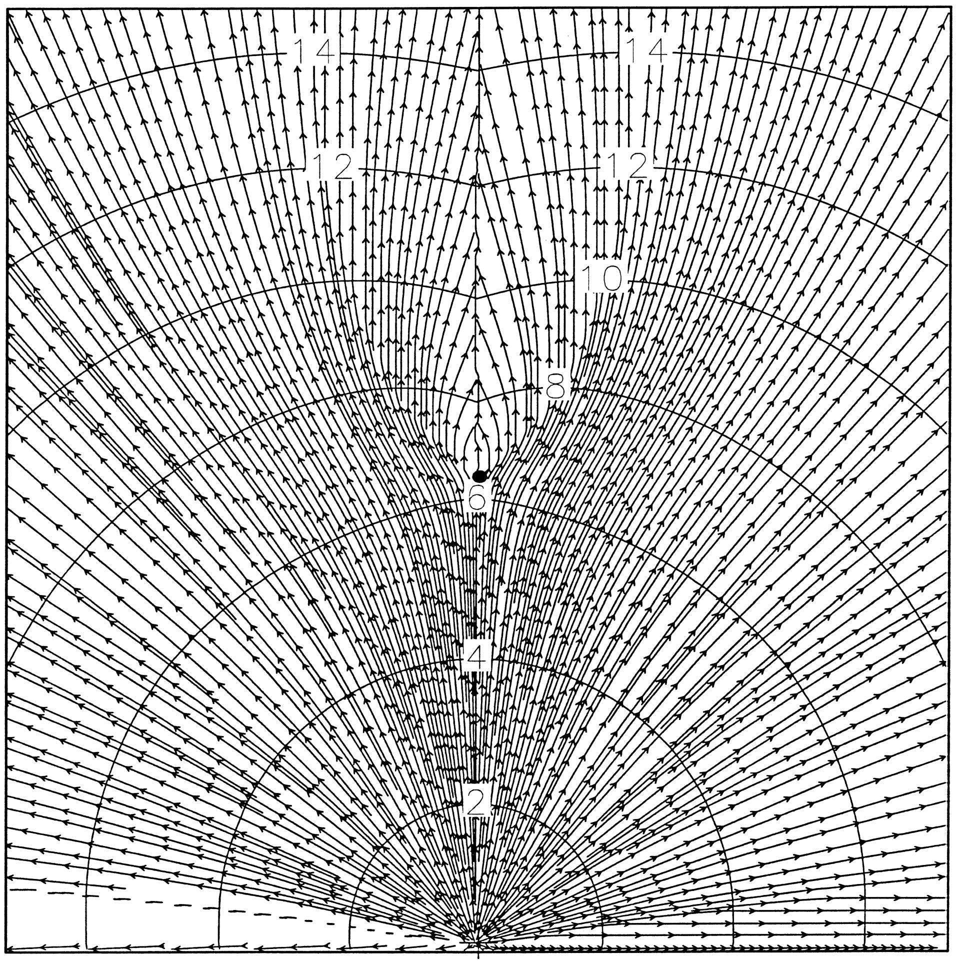

Figure 5. Demonstration of Finite Difference Path Planning for Drift in the Presence of a Sink (marked as filled circle). *Arrival times are marked on contours.*

5.2. *Stealth.* In stealth, the objective is to minimise the total detection risk in navigating a path to a target. The term *conspicuousness* is used as a composite of contributions from radar cross section, infra-red and electromagnetic emission. The risk of detection is given by the cumulative exposure:

$$C = \int_0^t c(\psi, \vec{X})\, dt, \tag{13}$$

where the journey duration is t, the instantaneous conspicuousness is c, and the vehicle position and heading are $\vec{X}$ and ψ.

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

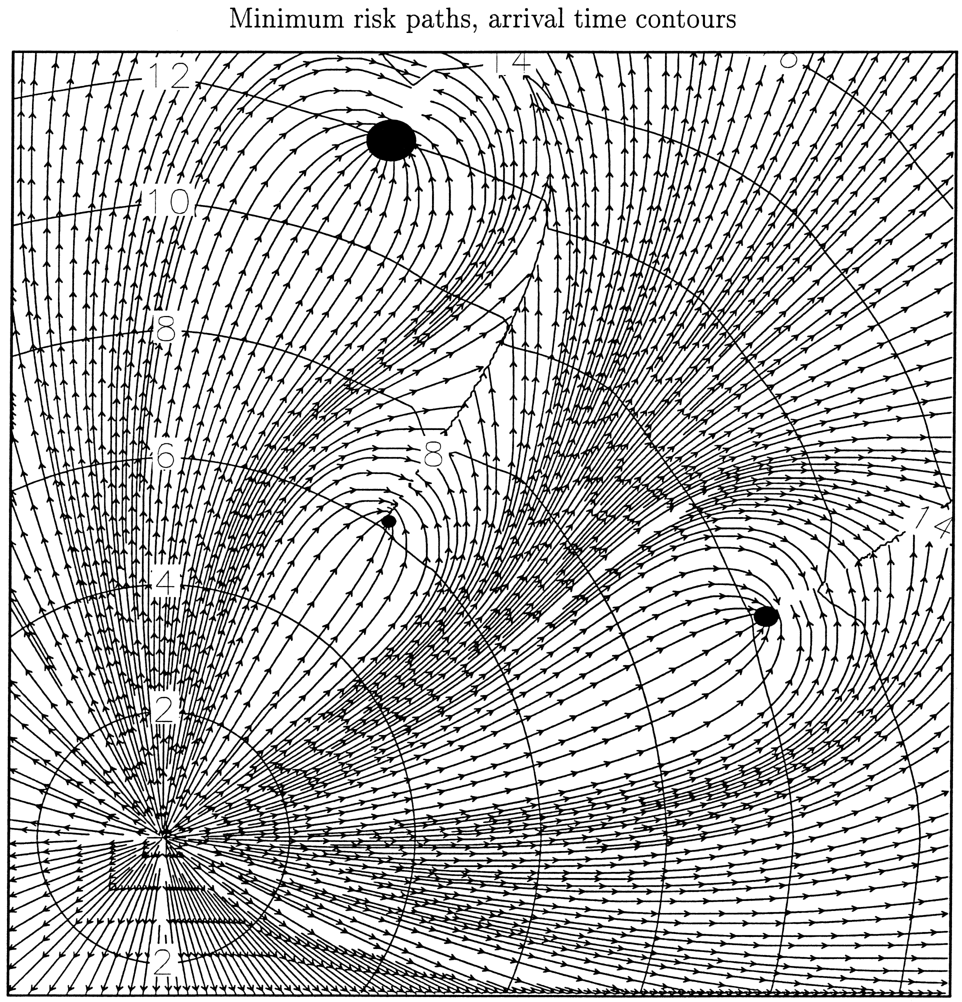

Figure 6. Demonstration of Finite Difference Path Planning for Stealth. *Three observer stations are marked by filled circles, size indicates detection power.*

The instantaneous conspicuousness is the sum of contributions from N detection stations:

$$\vec{\Delta}_i = \vec{X} - \vec{X}_i,$$
$$c = \sum_{i=1}^{N} C_i\, c_d(|\vec{\Delta}_i|)\, c_\psi(\psi - \angle \vec{\Delta}_i), \tag{14}$$

where C_i is the detection power and $\vec{X}_i$ is the position of station i. The functions c_d and c_ψ relate conspicuousness to relative distance and direction to a station.

Transformation of the minimum time property to minimum detection risk is achieved by a virtual slowness operator that is related to instantaneous conspicuousness. This approach would be very efficient at producing mission plans that indicate which targets can be reached with an acceptable level of risk. Conversely, using the FDPP in a conjugate form by planning from the desired goal, produces risk-

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

minimal trajectories for any location. This can be useful in determining the safest approach trajectories. This does not preclude the use of a realistic model for actual slowness calculations, so drift and resource management as well as stealthiness can be accounted for.

As an example of the application of FDPP to stealth, a domain with three observer stations is constructed with conspicuousness defined as

$$c_d = 90\,000/(|\Delta_i|^2 + 90), \tag{15}$$

$$c_\psi = 1{\cdot}0. \tag{16}$$

The observer stations have variable detection power of 0·003, 0·010, 0·005. The resulting optimal paths, isochrones and risk integrated over the paths are presented in Figures 6 and 7 for a field of 600 units square, and vehicle speed of 25 units.

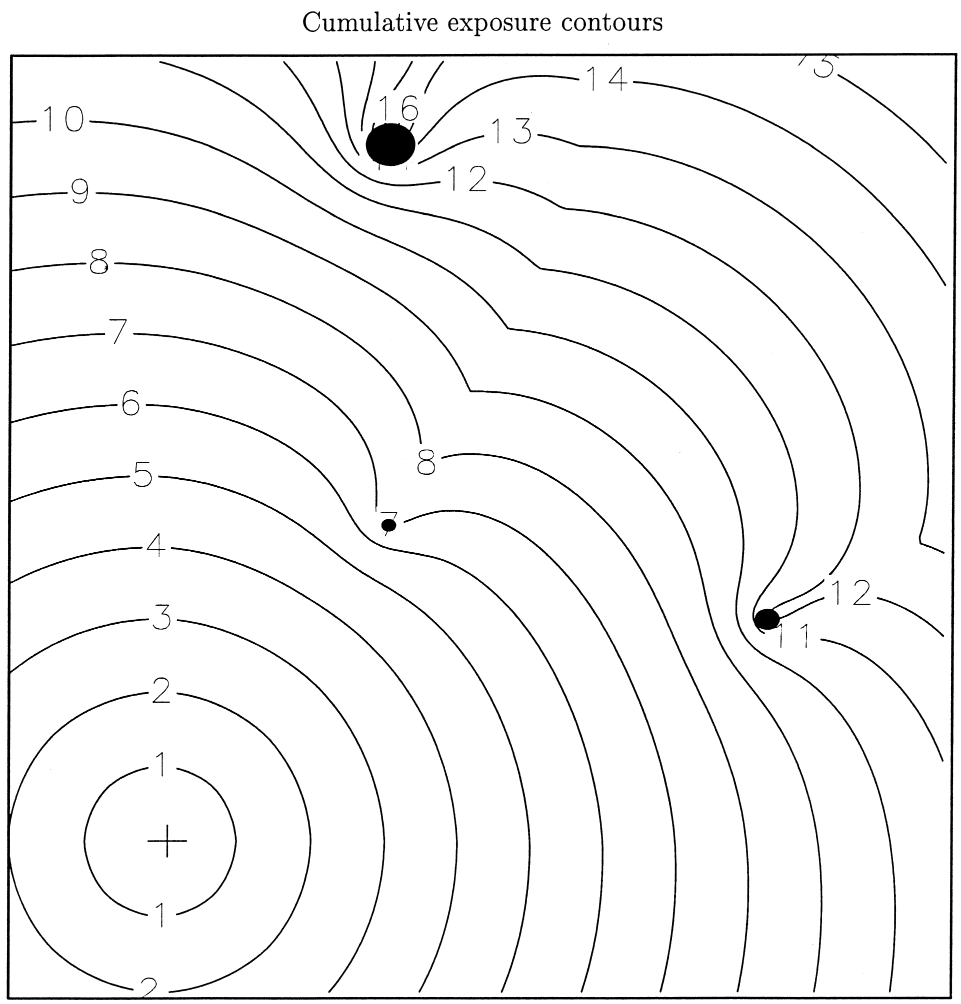

Figure 7. Demonstration of Finite Difference Path Planning for Stealth. *Three observer stations are marked by filled circles, size indicates detection power.*

https://doi.org/10.1017/S0373463399008334 Published online by Cambridge University Press

6. DISCUSSION. The wave propagation analogy has been previously realized. Sawyer described a graphical technique based on drawing a series of rings in which the radii are directly related to vehicle speed.[7] Bennett argued that this technique would be unsuitable for implementation on a computer on the basis that 'it is not known until the construction is complete which path will pass through the destination'.[8] From the work presented here, it would appear that his conclusion was too hasty – FDPP is competitive with graph-theoretic algorithms even when the destination is known *a priori*, and there are several benefits of planning for all free locations.

Bounds exist on the expected and worst-case running time and temporary storage required by the major graph-theoretic algorithms. The most efficient to date, a modified Dijkstra[9,10] algorithm, has an efficiency of near-$O(n)$; that is, as the number of nodes n increases, the running time per node increases. In the usual case where ring contractions are disabled, FDPP is strictly $O(n)$ and it provides all possible solutions.

If the destination is not known prior to the search, graph search methods would require $O(n^2)$ operations to plan paths to every node in the field. This is the worst-case scenario where the algorithm is applied afresh for every potential destination, and no doubt specialised versions of the algorithms could be developed that make use of previous search results. Nevertheless, the efficiency would be considerably worse than FDPP, which is still $O(n)$.

See Reference 6 for details of the accuracy of the FD approximation. The FD approach should be considered more accurate than a graph theoretic approach, on similar grids, since all arcs in the graph theoretic approaches have nodes as end-points, thereby compounding the discretisation error. With this said, the slowness queries will be subject to errors due to finite word length.

A further advantage of FDPP is that the memory required is linear with the number of nodes and, for a given mesh size, the execution time is constant. Finite difference methods lend themselves to neural network implementations since the calculations are based entirely on local (neighbouring) data. Thus it would be possible to construct a special purpose microprocessor for extremely fast planning. This sort of device would be quite useful in robotic vehicles and also manned ones. For instance, it could be used to provide a continuously updating map of reachable locations and expected arrival times, which would be useful in the event of engine/fuel emergencies, or as part of a flight management system employing 'free-flight', in which route planning is performed, to a large extent, on board the aircraft. In the military context, the stealth planner is an efficient way to implement safe corridor displays that were popularly advocated in the 1980s.

An advantage of point-to-field planning is that it does not pre-empt the goal position, which is useful where goals are mobile or depend on the path. Conversely, point-to-field approaches can be used for field-to-point planning, that is to find paths from all free locations to a single goal. In static environments, this eliminates the need for re-planning if the vehicle is momentarily side-tracked.

FDPP was initially developed as a guidance system for soaring assisted solar aircraft.[11] Great emphasis was placed on efficiency because computing power on such vehicles is severely limited. Moreover, multiple and redundant planning needs to be performed to account for uncertainty in weather forecasts. Point-to-field planning is useful in this application as it avoids wasted effort at planning for destinations that may be unreachable due to inclement weather. Secondly, it facilitates the selection of goals which maximise the average speed of the vehicle.

7. CONCLUSIONS. Recent burgeoning interest in robotic vehicles and free-flight navigation will demand high speed path planning algorithms. Finite Difference Path Planning is an interesting alternative to the graph search methods that currently dominate the field. Already, on single processor machines, FDPP offers the useful properties of a linear dependence of memory and execution time to the number of nodes on the discrete field. Secondly, the method would be highly suited to implementation as a massively parallel computation. Considering the ever-present uncertainty of the environment, particularly that of the weather, high-speed planning is likely to be an essential element in the practicality of automatic route planning.

Although FDPP is in its basic form intended to find minimum-time tracks, it has been demonstrated in this paper that the minimum-time property can be utilised for other objectives.

REFERENCES

[1] Pearl, J. (1994). *Heuristics*. Addison-Wesley.

[2] Bertsekas, D. (1991). *Linear Network Optimization: Algorithms and Codes*. MIT Press.

[3] Bertsekas, D. (1991). The auction algorithm for shortest paths. *SIAM Journal on Optimization*. Vol. **1**, pp. 425–447.

[4] Vidale, J. (1998). Finite-difference calculation of travel times. *Bulletin of the Seismological Society of America*. Vol. **78**, pp. 2062–2076, December 1988.

[5] Lecomte, I. (1993). Finite difference calculation of first traveltimes in anisotropic media. *International Journal of Geophysics*. Vol. **113**, pp. 318–342.

[6] Podvin, P. and Lecomte, I. (1991). Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools. *International Journal of Geophysics*. Vol. **105**, pp. 271–284.

[7] Sawyer, J. (1949). Theoretical aspects of pressure pattern flying. *Meteorological Report No. 3*, HMSO.

[8] Bennett, L. (1982). The development of programs for minimum-time tracks. This *Journal* **35**, 499–501.

[9] Moser, T. (1991). Shortest path calculation of seismic rays. *Geophysics*. Vol. **56**, pp. 59–67, January 1991.

[10] Johnson, D. (1977). Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*. Vol. **24**, pp. 1–13.

[11] Wharington, J. (1998). *Autonomous Control of Soaring Aircraft by Reinforcement Learning*. PhD thesis, Royal Melbourne Institute of Technology, (submitted).

KEY WORDS

1. Planning. 2. Automation. 3. Design.