

THE CONSTRUCTION OF DRAPE SURFACES WITH CONSTRAINED FIRST DERIVATIVES

RJ FOSSATI

**PO Box 1036, Southdale, 2135, South Africa
rfossati@fugroairborne.co.za**

and

JS WOLVAARDT

**Dept of Quantitative Management, Unisa
PO Box 392, Pretoria, 0003, South Africa
wolvajs@unisa.ac.za**

ABSTRACT

The need to construct optimal drape surfaces arises in airborne geophysical surveys where it is necessary to fly a safe distance above the ground and within the performance of the aircraft used, but as close as possible to the surface. The problem is formulated as an LP with constraints at every point of a grid covering the area concerned, yielding a huge problem.

The lifting algorithm is suggested. This is a surprisingly simple algorithm which starts with the drape surface at ground level and lifts it one point at a time. Only points which are too low relative to one or more of their neighbours are considered and they are lifted just enough to bring them into kilter with their neighbours. It is shown that the lifting algorithm is both exact and has great speed advantages. Some numerical results confirming exactness and speed are presented.

An enhanced method with better complexity is proposed and tested numerically.

1. INTRODUCTION

During airborne geophysical surveys measuring instruments of various kinds are transported across the area of interest, either on-board an aircraft or towed behind it. In order to obtain accurate readings the aircraft should fly **as close to the ground as possible**. Safety considerations dictate a minimum ground clearance of about 80 metres. Variations in ground topography prevent a fixed clearance from being maintained throughout a survey. For

example, a steep cliff requires the pilot to start his climb some distance away, thus causing the ground clearance to be larger at the base of the cliff than in more level areas.

Surveys are normally flown along parallel lines at a distance of between 50 and 2 000 metres apart. The first line might run from south to north along the western edge of the survey area. The next line then runs from north to south, displaced from the first by a distance of, say, 250 m. The third line runs from south to north again, at 500 m from the first line, and so on. Measurements are taken at regular intervals along the lines with the interval length typically between 5 and 10 metres. Once the entire area has been covered tie lines are flown in an orthogonal direction and with a somewhat coarser spacing.

In order to assess the feasibility of a given flight line, some height information about the surface is required. This is normally in the form of a digital terrain model (DTM), a set of values of the height above sea level of a grid of regularly spaced points. Such models may be quite coarse (eg a grid of points spaced 1 000 m apart) or quite fine (points 50 m or less apart). Obviously mathematically continuous functions do not exist for real-world topographies. It must be borne in mind that the spacing of the points at which measurements are taken and the spacing of the points of the DTM are not related at all. This paper is concerned with the construction of a lowest flyable surface on the DTM grid (with square cells). The flight lines are laid out on this surface *post facto*.

Aircraft performance during steep climb and steep descent varies considerably, particularly for the fixed-wing aircraft mostly used. The maximum rate of climb achievable usually is greater than the maximum rate of controlled descent (Pirrie). If a steep cliff ran normal to the flight lines and maximum rates of climb and descent were applied to all lines, the lines “up” the cliff would hug the flat surface at the base of the cliff more closely than the lines flown in the opposite direction, ie “down” the cliff. This would cause considerable height differences between adjacent lines, which would typically result in excessive levelling corrections having to be applied during processing of the data. Such corrections cause degradation, and in severe cases, loss of data, thus rendering the survey less effective. The smaller rate of altitude change is therefore used.

An obvious solution to the stated problem would be to construct a flat flight surface lying high enough above the point of highest elevation in the survey area so that the aircraft’s safety

would not be endangered. This method has the drawback that low-lying areas are much farther away from the measuring instruments than higher-lying ones, and these readings would suffer from a lower resolution. Consequently this method is an option only if the survey area has little topographical variation.

A better solution is to construct a drape surface over the survey area, which **hugs the ground as closely as possible within the constraints of the aircraft performance parameters**. At no point may it be closer to the ground than the minimum safety height, and its shape along any vertical section must be readily realisable by the aircraft to be used in the direction in which it is flown.

Note that the drape surface is constructed first, then flight and tie lines are flown on the drape surface and measurements taken. If the aircraft flies at the height of the drape surface excessive levelling corrections are not needed, altitude changes are within the capability of the plane, the plane flies at a safe altitude, and the quality of the resulting measurements is very good although not necessarily maximised. Since the actual total ground clearance summed at the points of measurement is the true value of the quality of the observations, this value can be suboptimal. The drape surface is based on all the points in the square grid of the DTM and may take into account higher points which are missed by the different grid on which the measurements are finally taken. (The flight lines are quite far apart in terms of the DTM grid.) The possible advantage of a better sum of clearances at the points of measurement only, is outweighed by the safety of taking into account all points in the DTM. In this regard it should be borne in mind that even excellent pilots average a deviation of about a metre from an ideal line.

Pre-constructing a drape surface provides a surface that is **best** at “hugging” the geographical surface and is **always flyable**, regardless of where the flight lines are placed upon it.

This paper examines the method of constructing such a drape surface with the objective of minimizing the total ground clearance.

2. FORMULATION AS A LINEAR PROGRAM

Consider a survey area A consisting of a grid of I rows and J columns. Let $d_{i,j}$ be the topographical height of the point (i, j) , where $i = 1, \dots, I$ and $j = 1, \dots, J$. Let $z_{i,j}$ be the height of the drape surface at the same point. Since both the DTM and the drape surface are discrete, the slope is approximated by the difference in height of two neighbouring points rather than the continuous derivative. The slope constraints between the point (i, j) and its four neighbours can then be written as

$$\begin{aligned} z_{i,j} - z_{i+1,j} &\leq \alpha & z_{i+1,j} - z_{i,j} &\leq \alpha \\ z_{i,j} - z_{i,j+1} &\leq \alpha & z_{i,j+1} - z_{i,j} &\leq \alpha \\ z_{i,j} - z_{i-1,j} &\leq \alpha & z_{i-1,j} - z_{i,j} &\leq \alpha \\ z_{i,j} - z_{i,j-1} &\leq \alpha & z_{i,j-1} - z_{i,j} &\leq \alpha \end{aligned}$$

where α is the maximum height differential between adjacent points derived from the maximum drape surface slope and the step length of the grid.

Since every point on the drape surface must be at least the minimum flying height, m , above the topographical surface

$$z_{i,j} \geq d_{i,j} + m \quad \forall i, j.$$

The objective is to create as “low” a drape surface as possible. We therefore want to

$$\min \sum_{i,j} (z_{i,j} - d_{i,j} - m).$$

Examination of the constraints relating to the slope shows that each constraint actually occurs twice for any point (i, j) not on the edge of the area. The number of constraints can therefore be reduced by considering only the differential slope to the points north and east of any point under consideration. Hence only the first four constraints of the eight above are required.

The objective function

$$\min \sum_{i,j} (z_{i,j} - d_{i,j} - m)$$

can be simplified to

$$\min \sum z_{i,j}$$

since both $\sum d_{i,j}$ and m are constant. In the rest of this article m is disregarded; its effect is merely additive.

The linear program can now be written in the following standard form with \leq constraints and a max objective function:

$$\text{Max } (- \sum z_{i,j})$$

subject to

$$z_{i,j} - z_{i+1,j} \leq \alpha \quad i = 1, \dots, I-1, \quad j = 1, \dots, J$$

$$z_{i+1,j} - z_{i,j} \leq \alpha \quad i = 1, \dots, I-1, \quad j = 1, \dots, J$$

$$z_{i,j} - z_{i,j+1} \leq \alpha \quad i = 1, \dots, I, \quad j = 1, \dots, J-1$$

$$z_{i,j+1} - z_{i,j} \leq \alpha \quad i = 1, \dots, I, \quad j = 1, \dots, J-1$$

and

$$-z_{i,j} \leq -d_{i,j} \quad i = 1, \dots, I, \quad j = 1, \dots, J$$

with

$$\text{all variables } \geq 0.$$

For clarity the first four sets will be called the “slope constraints”, while the last set constitutes the “height constraints”.

The size of the problem

For a survey area covered by a rectangular grid of I rows and J columns we have $J(I-1) + J(I-1) + I(J-1) + I(J-1)$ slope constraints and IJ height constraints giving a total of $5IJ - 2(I+J)$ constraints.

A typical grid of, say, 1 000 rows by 1 000 columns, would require 1 000 000 variables and 4 996 000 constraints. Solution by the simplex method is simply not possible. An efficiently packed matrix would still have to account for some 9 000 000 non-zero entries and the bookkeeping to access these values. The problem, however, lies with the large number of constraints since the time for the solution of an LP problem is typically proportional to this (Chvátal, p. 45). Clearly another method must be found if this problem is to be solved in practice.

3. THE LIFTING ALGORITHM

We associate a state variable with each point in the grid. This takes on one of three states: *untouched*, *set* and *fathomed*. A point in the *untouched* state has a drap height equal to the topographic height and its influence on neighbouring points has not yet been evaluated. A point in the *set* state has its final drap height, but it has not yet been used to adjust the values of other points. A point in the *fathomed* state has its final drap height set and its influence on neighbours has been fully fathomed.

Step 1: Set $z_{i,j} = d_{i,j}$ for all points (i, j) in the grid. This implies that all the height constraints will be satisfied.

Step 2: Set the state of all points (i, j) to *untouched*, except for the highest point in the grid, for which the state is set to *set*. It also becomes the first *current* point.

Step 3: Examine all the *untouched* neighbours of the *current* point. If it is lower than α from the *current* point, raise it so that the height differential between it and the *current* point is exactly α . Do not adjust any neighbours of the *current* point whose height differs from that of the *current* point by less than α . All neighbours now move into the *set* state, while the *current* point becomes *fathomed*.

Step 4: Find the highest point (based on the current z -value) which is either *untouched* or *set*. This is the new *current* point. Go to step 3. If all points have been *fathomed* the algorithm terminates (in the next step).

Step 5: The drape surface has been determined. The objective function value is taken to be the sum of the elevations of all the points of the drape surface.

In practice the drape surface is constructed in regions which emanate from the peaks of the topographic surface, each of which forms a nucleus from which a region of primal feasibility spreads until the entire drape surface is feasible. Once the final drape surface has been constructed, every point on it is either at its original height, or it has been raised and it maintains a maximum-slope relationship with a higher-lying neighbour.

Example

The following small example (using only whole numbers for simplicity) may serve to illustrate the lifting algorithm. The parameter α has been set to 2. The following matrix shows the tiny topographical 3 x 4 matrix for which a drape surface is to be constructed.

7	10	13	6
4	10	12	7
11	8	6	14

In the following diagrams the set state is indicated by an asterisk while the fathomed state is shown with two asterisks. Unmarked cells are untouched. We begin with all points untouched, except for the highest, (3;4), which is set.

7	10	13	6
4	10	12	7
11	8	6	14*

The current point is (3;4). Consider its two neighbours [(3;3) and (2;4)]. Both are untouched and have an unacceptable height differential with the current point. Hence they are both raised to a height α less than that of point (3;4), ie 12. Point (3;4) has now been fathomed, while points (3;3) and (4;2) move into the set state.

7	10	13	6
4	10	12	12*
11	8	12*	14**

The highest unfathomed point is now (1;3) with a z -value of 13 and it becomes the current point. Consider its neighbours [(1;2), (1;4) and (2;3)] which are all untouched. The height difference between (1;3) and (2;3) is acceptable, but with the other two neighbours it is not. Raise points (1;2) and (1;4) to 11. Point (1;3) becomes fathomed, points (1;2), (1;4) and (2;3) become set.

7	11*	13**	11*
4	10	12*	12*
11	8	12*	14**

The method continues in this fashion until all points have been fathomed. The figure below shows the final result.

9**	11**	13**	11**
9**	10**	12**	12**
11**	10**	12**	14**

Optimality

It can be shown that the drupe surface generated by the lifting algorithm is both feasible and optimal. For this purpose consider the points (i, j) in the area under consideration and the heights $z_{i,j}$ that are allocated to them by the lifting algorithm.

Feasibility requires that both the height constraints and the slope constraints are satisfied. Since the starting values are all set to $d_{i,j}$ and can only be increased, the height constraints ($z_{i,j} \geq d_{i,j}$) are satisfied. The slope constraints can be represented by the requirement that no point has a lower neighbour with height difference exceeding α . Since the algorithm eliminates this possibility for a current point before marking it fathomed and stops only when all points have been fathomed, this is also satisfied.

To prove optimality it is first shown that each point on the drape surface is either a member of set A or set B where $A = \{(i, j) | z_{i,j} = d_{i,j}\}$ and $B = \{(i, j) | z_{i,j} > d_{i,j}; z_{i,j}(\alpha)\}$. Here $z_{i,j}(\alpha)$ indicates that (i, j) is in a binding relationship with at least one of its neighbours, meaning that the height of that neighbour is exactly α larger than $z_{i,j}$.

The lifting algorithm starts with all points in A . It lifts a point only if it has a neighbour the height of which exceeds its own by α or more, and then lifting it to be in a binding relationship with that point. This moves the point from set A to set B . This happens while the neighbour is the current point. When the neighbour becomes fathomed the point becomes set and since the height of neither can be changed in later steps of the lifting algorithm it holds for the final drape surface. This means that once a point becomes an element of B it remains one. Consequently for the final $z_{i,j}$ values, every point is either in A or B .

Assume the surface is not optimal. Then there must exist a point (i, j) whose height $z_{i,j}$ can be lowered without sacrificing feasibility. This is only possible if it is both higher than its topographic height $d_{i,j}$ and not in a binding relationship with any higher neighbour (if a fixed substructure can be lowered this will imply the existence of such a point at its edge) and therefore (i, j) does not belong to A or B . This contradicts the fact that all (i, j) belong to A or B . The assumption is wrong. The surface is optimal.

Dual method

It can be shown that the lifting algorithm is in fact a dual algorithm. The initial infeasible primal solution can be matched with a feasible dual that is in a complementary slackness relation to it. It can further be shown (this is complicated and space consuming resulting in these few lines on duality) that each lifting operation is equivalent to a dual iteration. The lifting algorithm stops when primal feasibility is reached. Since dual feasibility and complementary slackness are maintained throughout, this solution is optimal.

Numerical confirmation

In order to numerically confirm the correctness of the drape surfaces generated using the lifting algorithm, it was coded and its results compared to that of commercially available software using the standard simplex method for solving linear programming problems. The commercial software library selected was XA by Sunset Software Technology, San Marino, CA, USA; this suite has remarkable abilities for handling sparse matrices and displays an excellent speed in solving LP problems. The table below shows the results obtained for some (smaller) problems.

Problem Size		Optimum using	Optimum using
Rows x columns	No. points (n)	XA (standard LP)	Lifting algorithm
4 x 5	20	183,0	183,0
16 x 16	256	4 091,48	4 091,33
32 x 32	1 024	16 487,3	16 487,1
64 x 64	4 096	65 757,1	65 757,2
100 x 100	10 000	161 232	161 233
200 x 50	10 000	161 023	161 024
128 x 128	16 384	263 526	263 529
100 x 200	20 000	321 635	321 639
200 x 200	40 000	643 406	643 414

As was to be expected, the answers obtained using the lifting algorithm are the same as those obtained using an exact “traditional” algorithm. There are, of course, some small variations between the answers. These are rounding errors but they are to be expected, considering the often very large number of pivot operations performed by the LP software and the limited accuracy (about six decimal digits) in the usual representation of floating point numbers.

Complexity

Although the lifting algorithm is a single-pass method, at every iteration the highest unfathomed point in the grid must be found. The complexity of the method is thus $O(n^2)$ rather than $O(n)$ where n is the number of points in the grid.

This result is numerically confirmed in a later table.

However, it is possible to achieve the same primal feasible solution in a quicker way.

4. ENHANCING THE LIFTING ALGORITHM

The enhanced lifting algorithm

Step 1: Set $z_{i,j} = d_{i,j}$ for all points (i, j) in the grid. This implies that all the height constraints will be satisfied.

Step 2: Begin with the top left point as the current point – here the point $(1;1)$.

Step 3: Examine the height difference between the current point and its right-hand neighbour, ie $(i, j + 1)$. If there is no right-hand neighbour, ignore the rest of this step. If the height difference between the current point and the neighbour exceeds α raise the lower of the two points, so that the difference in question becomes exactly α .

Step 4: Repeat the process of step 3 with the bottom neighbour of the current point, ie $(i + 1, j)$.

Step 5: Move the current point one to the right. If the current point is the last one of the current line, move the current point to the left-most point of the next line. If the current point is the bottom right point of the grid, move on to step 6. Otherwise repeat the procedure for the current point from step 3 onwards.

Step 6: The forward pass has now been completed. Set the current point to the bottom right point in the grid, ie (I, J) .

Step 7: Examine the height difference between the current point and its left-hand neighbour, ie $(i, j - 1)$. If there is no left-hand neighbour, ignore the rest of this step. If the height difference between the current point and the neighbour exceeds α raise the lower of the two points, so that the difference in question becomes exactly α .

Step 8: Repeat the process of step 7 with the top neighbour of the current point, ie $(i - 1, j)$.

Step 9: Move the current point one to the left. If the current point is the first one of the current line, move the current point to the right-most point of the previous line. If the current point is the top left point of the grid, move on to step 10. Otherwise repeat the procedure for the current point from step 7 onwards.

Step 10: The reverse pass has now been completed and the drape surface has been determined. The objective function value is taken to be the sum of the elevations of all the points of the drape surface.

This procedure raises the areas to the right of and below peaks in the original topographic surface during the forward pass. During the reverse pass areas to the left of and above peaks are also raised into pyramid-shaped sections of the drape surface. The result which is obtained is exactly the same as that which results from the lifting algorithm.

Example

The enhanced method is illustrated here with the same example as the one used for the lifting algorithm. The initial z -grid (same as the topographical grid) is shown again below for convenience. Again the parameter $\alpha = 2$ is used.

7	10	13	6
4	10	12	7
11	8	6	14

The forward pass begins at the top left point. Observe that $z_{1,2} - z_{1,1}$ exceeds α . Therefore point (1;1), the lower of the two, is raised so that $z_{1,2} - z_{1,1} = \alpha$. The z -values are now:

8	10	13	6
4	10	12	7
11	8	6	14

However, work on the first point is not finished yet. The height difference between the top left point and the one below it is too large ($z_{1,1} - z_{2,1} > \alpha$), thus point (2;1) must be raised. This brings us to the following surface:

8	10	13	6
6	10	12	7
11	8	6	14

The next point to be considered is (1;2) which currently has $z = 10$. At this point the difference between (1;2) and (1;3) must be adjusted, and (1;2) is raised by one unit:

8	11	13	6
6	10	12	7
11	8	6	14

Notice how this, in effect, introduces an infeasibility again with respect to the height difference between (1;1) and (1;2). This infeasibility will be removed on the reverse pass.

The height differential between (1;2) and (2;2) is in order. The method therefore proceeds to point (1;3). Here the height difference with its right-hand neighbour exceeds α and point (1;4) must be raised. Once the examination of point (1;4) has been completed, the method moves on to the next row and examines point (2;1).

At the end of the forward pass the grid's z -values are as follows:

8	11	13	11
9	10	12	12
11	9	12	14

The reverse pass begins by considering point (3;4). The height differences to its left neighbour (3;3) and its upper neighbour (2;4) are fine, so the method moves on to the next point in reverse reading order; this is point (3;3).

At point (3;3) the height difference between the point and its left-hand neighbour is too large, and point (3;2) must therefore be raised:

8	11	13	11
9	10	12	12
11	10	12	14

Continuing in this manner we observe that no further adjustments need to be made except at the very end. At this step point (1;2) is under consideration and we find that the height difference between (2;1) and (1;1) (its left neighbour) exceeds α . Therefore point (1;1) is raised and the algorithm terminates with the solution shown below. This solution is identical to that generated by the lifting algorithm.

9	11	13	11
9	10	12	12
11	10	12	14

Optimality

Once again it can be shown that the drape surface generated by the enhanced lifting algorithm is both feasible and optimal. Again consider the points (i, j) in the area under consideration and the heights $z_{i,j}$ that are allocated to them by the enhanced lifting algorithm.

The optimality of the enhanced lifting algorithm follows by the same proof as used for the normal lifting algorithm.

Feasibility again requires that both the height constraints and the slope constraints are satisfied. Since the starting values are all set to $d_{i,j}$ and are only ever increased, the height constraints ($z_{i,j} \geq d_{i,j}$) are satisfied.

The slope constraints require that the height difference between a point (i, j) and its four neighbours is no greater than α . During the forward pass each point either remains at its topographic height or becomes part of a chain of binding relations to a local peak which lies

to its west or north. Hence, after the forward pass there cannot be any point (i, j) for which $z_{i,j} - z_{i,j+1} > \alpha$ or $z_{i,j} - z_{i+1,j} > \alpha$. Similarly, after the reverse pass there cannot be any point (i, j) for which $z_{i,j} - z_{i,j-1} > \alpha$ or $z_{i,j} - z_{i-1,j} > \alpha$. The slope constraints are also satisfied.

The drape surface generated by the enhanced method is feasible and optimal.

Complexity

The enhanced lifting algorithm is also a single-pass method but every point is visited exactly twice. The complexity of the method is thus $O(n)$ where n is the number of points in the grid. This improves on the complexity of $O(n^2)$ of the original lifting algorithm.

This result is numerically confirmed in a later table.

Numerical confirmation

Optimality has been numerically confirmed by comparing the results of the enhanced with that of the ordinary lifting algorithm for a large number of problems. The results were identical.

5. NUMERICAL RESULTS

Both the lifting algorithm and the enhanced method were coded and run on a variety of examples. The table below gives execution times in seconds for the examples. Since the wall clock timings of the enhanced method were less than one second in all cases the method was slowed down programmatically to obtain more distinctive readings. The more or less constant terms in the columns n^2/t_1 and n/t_2 demonstrate the complexities of $O(n^2)$ and $O(n)$ for the two methods respectively. The three ratios indicated by dashes are not shown because of the dominance of measuring errors. All calculations were done on a 233 MHz Pentium.

Problem size		Lifting algorithm		Enhanced method	
Rows x columns	No. points (n)	Time (t_1) (s)	n^2/t_1	Time (t_2) (s)	n/t_2
4 x 5	20	0		0	
16 x 16	256	0		1	-

32 x 32	1 024	1	-	2	-
64 x 64	4 096	7	$2,4 \times 10^6$	9	455
100 x 100	10 000	41	$2,4 \times 10^6$	21	476
200 x 50	10 000	41	$2,4 \times 10^6$	21	476
128 x 128	16 384	115	$2,3 \times 10^6$	34	482
100 x 200	20 000	173	$2,3 \times 10^6$	40	500
200 x 200	40 000	709	$2,3 \times 10^6$	81	493

It is of interest to note that the real (unslowed) execution time of the enhanced method for all the problems above was less than one second. The execution speed for XA varied from a very comparable less than 1 second and 5 iterations for the 4 x 5 problem to a rather less acceptable 6580 seconds (about 1 hour 50 min) and 34847 iterations for the 200 x 200 problem. Clearly, the enhanced method comes into its own as the problem size increases.

Further research

Not very much more need or can be said about the drape problem with first derivative constraints. It is hardly conceivable that a solution complexity of less than linear in the problem size can be found.

However, there is one aspect of the solution which does not mirror the practical situation very well: the presence of sharp peaks and troughs where the derivative changes from one extreme to another. In the practical case of drape flying these would have to be rounded out. This implies including constraints on the second derivatives. An in-depth treatment is in preparation.

REFERENCES

- [1] V. CHVÁTAL, *Linear Programming*, W.H. Freeman and Company, New York (1980).
- [2] S.J. PIRRIE, Private communication relating to fixed-wing aircraft parameters (1999).

ACKNOWLEDGEMENTS

Thanks are due to the people at Geodass (Pty) Ltd, now Fugro Airborne Surveys (Pty) Ltd, for their support and assistance in this work. Thanks are also due to the Department of Quantitative Management of Unisa for providing the XA linear programming library. The authors would also like to thank the referees of this article for their input.