A DIRECT SEARCH PROCEDURE

The purpose of this note is to present an algorithm for finding the maximum of a function of two variables. The aim is to step systematically through the parameter space in search of the maximum in a manner that is rapid and efficient and which avoids unnecessary function evaluations.

The algorithm can be applied, for example, to the task of finding the maximum-likelihood estimates of the parameters of a 2nd-order autoregressive AR(2) model which can be represented by the equation

$$\begin{aligned}\varepsilon(t) &= (1 - \alpha_1 L + \alpha_2 L^2)y(t) \\ &= \{1 - 2\rho\cos(\omega) + \rho^2 L\}y(t).\end{aligned} \tag{1}$$

The likelihood can be expressed as a function of the parameters $\rho \in [0, 1)$ and $\omega \in [0, \pi]$. Therefore, the search for the maximum point can be confined to a rectangular region in the parameter space.

Imagine that we are located at a point, described as the base point, which is at the centre of one of the cells of a rectangular mesh that has been cast over the parameter space. The immediate objective is to attain a higher function value by stepping from the base point to one of the neighbouring points that lie in the adjacent cells.

If no such improvement in the function value is available, then we must confine our search to the cell containing the base point. If one of the neighbouring points does offer an improvement in the function value, then we may attempt a further improvement by taking another step in the same direction to a further point.

If the further point does not offer an improvement, then we must look for an improvement elsewhere in the vicinity of the neighbouring point that did offer an improvement. In doing so, we must avoid revisiting any point that we have already examined; and this requires that we keep track of our movements.

NW		*N*		*NE*
	14	2	6	
W	7	1	3	*E*
	35	5	15	
SW		*S*		*SE*

The figure above depicts a part of the mesh surrounding the base point, which is in the cell labelled by 1. The adjacent cells that lie in the cardinal directions to the north, east, south and west are labelled with 2, 3, 5 and 7, which are prime numbers, whereas those that lie in directions diagonal from the base point are labelled with products of the prime numbers in the two cells that are nearest. This numbering system serves to identify the direction of the

movement to the current base point from the previous base point, recorded in the variable *oldDirection*, and the direction of a movement away from it, recorded in the variable *direction*.

The base point has been reached by passing through one of the adjacent cells in which the function value of corresponding point has been assessed and found to be less than that of the base point. If the direction in moving to the base point has been a cardinal one, then *oldDirection* will have the value of one of the constants $north = 2$, $east = 3$, $south = 5$ or $west = 7$. If the direction has been a diagonal one, then *oldDirectsion* will take the value of the product of two such constants, e.g. $southWest = south * west = 35$.

Starting at the base point, we investigate whether any improvements are available by trying the cell above or the cell below in; and we make a move or we stay put accordingly. The direction of a move is recorded in the variable *direction*. The decision to remain within the base cell is recorded by setting $direction := stayPut = 1$.

In assessing a potential move along the north–south axis, it may be possible to avoid one of the function evaluations. If the previous move to the base point was from the south, which would be indicated by $oldDirection = north$, then there is no need to look to the south. Similarly, if the move was from the north, there is no need to look north.

The next step is to investigate whether any improvements are available by moving to the right or the left along the east–west axis. First, we consider looking east. If the cell question has already been visited, then it can afford no improvement and *dontLookEast* is *true*. The following rules may be applied which take account both of the move to the base point, recorded in *oldDirection* and of the recent move, if any, along the north–south axis, recorded in *direction*:

if $(oldDirection = west)$ **and** $(direction = stayPut)$ **then** *dontLookEast*;

if $(oldDirection = southWest)$ **and** $(direction = north)$ **then** *dontLookEast*;

if $(oldDirection = northWest)$ **and** $(direction = south)$ **then** *dontLookEast*;

Observe that, if we have just completed a move along the north–south axis, then $direction$ **in** $[north, south]$, in which case

$(oldDirection$ **in** $[southWest, northWest])$

is equivalent to

$(oldDirection$ **div** $direction) = west$.

Therefore, the rules for avoiding revisiting the cells can be expressed equivalently as

if $((direction = stayPut)$ **and** $(oldDirection <> west))$ **then** *lookEast*

if $((direction <> stayPut)$ **and** $(oldDirection$ **div** $direction <> west))$ **then** *lookEast*

If an improvement is available, then a move to the east is appropriate, and, having made it, we may set $direction := direction * east$. Otherwise, we should consider a move to the west. If we have not moved to the east already, then we should look to the west. We may do so on the condition that

$$direction \textbf{ not in } [east,\ northEast,\ northWest].$$

This is equivalent to the condition

$$(direction \textbf{ mod } east) <> 0,$$

which is a more elegant expression.

In looking west, we can avoid revisiting the cells that have been associated with previous function evaluations. This can be achieved by applying rules that are equivalent to those that would accompany an exploration to the east.

Once we have determined to move away from the base point in one direction or another, we must investigate whether a further move in the same direction is justified that would carry us into one of the cells labelled *N*, *NE*, *E*, *SE*, *S*, *SW*, *W* or *NW*. Such a move would provide a base point from which a new search can commence that runs no danger of revisiting any of the points.

If no further move is justified, then we must take additional care to avoid revisiting old points in the next round of searching. In the first place, if $direction$ **in** $[north, south]$, i.e. if the direction taken in this round has been a cardinal one, then it is clear that we will have already investigated the cells that lie to the north and to the south of the current cell, as well as those that lie to the east and west; and it is appropriate to stay put within the current cell.

In staying put, we must reduce the size of the mesh to ensure that the current cell contains at least 25 sub-cells. This implies reducing the step length by a factor of 5 at least. In fact, it is common to reduce the step length by a factor of 10.

If there has been no further move and if the variable *direction* has an diagonal value, then the current cell will have been reached via two successive moves—the first being along the north–south vertical axis, and the second being along the east–west horizontal axis. We must to avoid revisiting the point from which a move has just been made or revisiting its predecessor by backtracking along the east-west axis. For this purpose, it is appropriate to replace the diagonal value of the *direction* variable by the appropriate cardinal value, *east* or *west*, so as to to record the direction of the most recent step. Thus

$$\textbf{if } (direction \textbf{ mod } east = 0) \textbf{ then } direction := east;$$

and

$$\textbf{if } (direction \textbf{ mod } west = 0) \textbf{ then } direction := west.$$

The current point will become the base point and the value of *direction* will be assigned to *oldDirection* in preparation for the next round.