Line Generalisation Algorithms – Specific Theory

Digital Generalisation

Digital generalisation can be defined as “the process of deriving, from a data source, a symbolically or digitally-encoded cartographic data set through the application of spatial and attribute transformations”. The generalisation of digital spatial data is done for various reasons including:

- ✍✍to minimise the amount of data to be processed;
- ✍✍to filter out errors and consolidate trends;
- ✍✍to produce maps at a variety of scales (from the one source); and
- ✍✍to remove and graphically modify features for effective cartographic communication and aesthetics.

The many forms of generalisation that can be applied to vector-format data are grouped into the following five categories:

point
line
polygon (area)
volume
holistic feature.

There are a variety of different algorithms available in GIS to handle all of these processes, however for the purposes of this module we are only interested in the basic techniques available for **line generalisation**.

In general, up to 80% of the information on a digital map consists of lines, giving rise to the extensive list of generalisation operators that exist - including simplify, smooth, displace, merge, enhance and omit. For simplicity and clarity, this module focuses on two line generalisation operators: **Line Simplification** and **Line Smoothing**. For performing each of these operations there are numerous generalisation algorithms in use, however this module will only cover two in detail (one for each of the operators).

Each of the links below will take you to a page comprising the background theory for that particular line generalisation operator and also links to the relevant interactive lessons (which can also be accessed from the main menu).

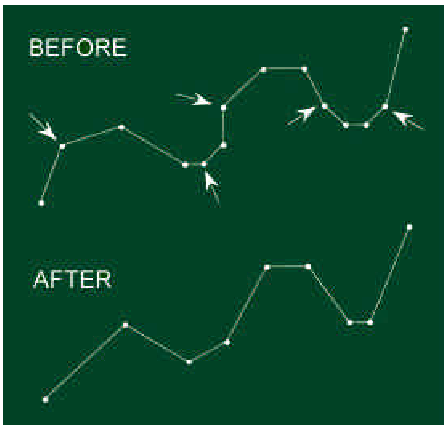
It doesn't matter whether you begin with Line Simplification or Line Smoothing, as all theory pages are fully inter-linked. The 'download' link at the bottom of this page will allow you to save all of the theory you are presented with in this module.

Line Simplification   Line Smoothing

Line Simplification Algorithms

Line simplification algorithms have been developed over the years for the purpose of 'weeding out' redundant or unnecessary coordinate information from line features, whilst retaining the perceptual characteristics of the line. They generally work via the application of some geometric criterion to a line's coordinate pairs (such as distance between points or displacement from a centre-line). Figure 1 below shows a line before and after it has been simplified, with arrows representing the points eliminated during the process.

Figure 1. Line Simplification (Source: McMaster & Shea 1992).



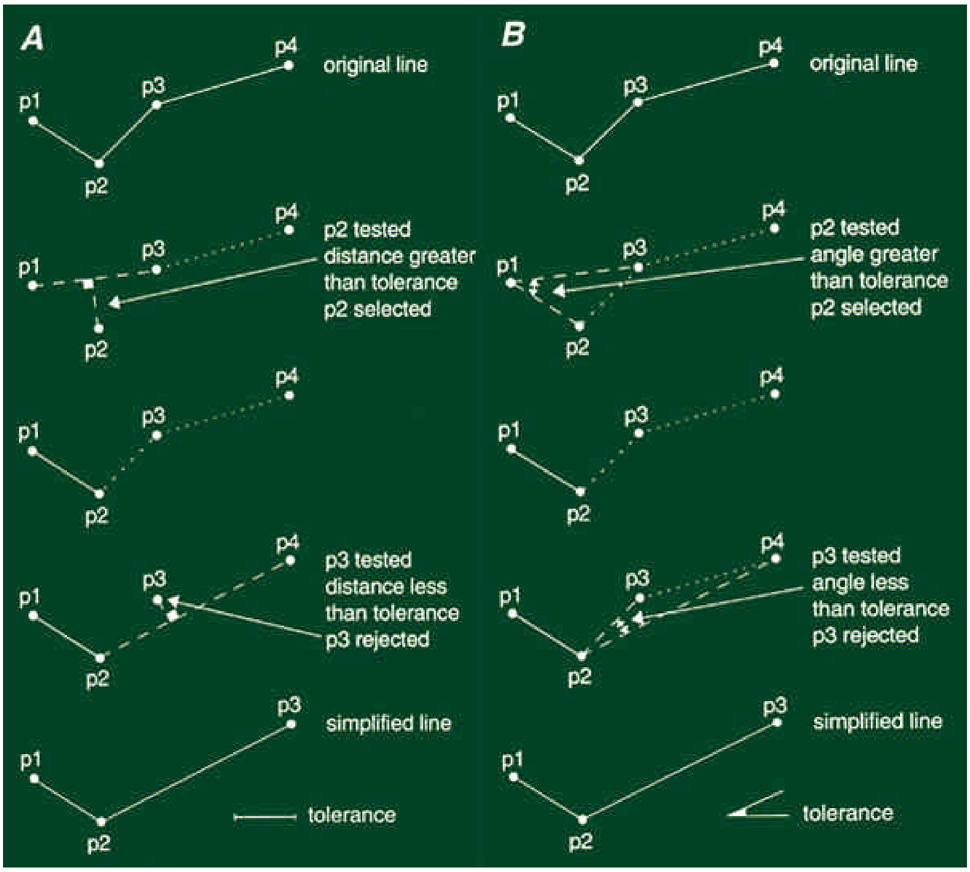
There are a series of justifications for the simplification of digitised lines. These are summarised below:

1. **Reduced plotting time:** simplification reduces the number of (x,y) coordinate pairs, increasing the plotting speed (a faster pen speed also improves the aesthetic qualities of the line).
2. **Reduced storage space:** simplification can reduce a large data set by up to 70%, decreasing both the amount of storage space required for the data and the cost of storing it.
3. **Faster vector to raster conversion:** a simplified coordinate set will enable faster conversion from vector to raster mode.
4. **Faster vector processing:** the time needed for processing vectors, including translation, rotation, re-scaling, cartometric analysis and some symbol-generation is greatly reduced with a simplified data set.

Local Processing Routines

As a basic introduction to line simplification algorithms, Figure 2 illustrates two local processing routines. The perpendicular distance algorithm is displayed on the left (***A***) and works by sequential operations on a triad of coordinate pairs. Firstly, a perpendicular is constructed to the vector connecting the first and third points (p1 and p3) of the triad, from the intermediate point (p2). If this distance is larger than the user-defined tolerance, the intermediate point is retained, however if it is shorter than the tolerance, the point is eliminated. The next three points (p2, p3 and p4) then form the new triad and the process is repeated. This is done until the end of the line is reached.

Figure 2. Local Processing Simplifiers (Source: McMaster & Shea 1992).



(Not to scale)

The angular tolerance algorithm is displayed on the right of the figure (***B***) and works via a similar process to that described above, only with an angular tolerance as measured between vectors connecting points p1 and p3, and p1 and p2.

The two algorithms presented in Figure 2 and described above form the basis of most other line simplification algorithms. The algorithm at the focus of the line simplification section of the module is known as the Lang Simplification Algorithm – an extended version of the perpendicular distance algorithm in Figure 2***A***. The Lang Simplification Algorithm is credited, among other things, with being able to maintain the original geometric characteristics of a line. There are links below (on the right) to two interactive lessons which will help you to visualise how Lang's algorithm is applied when simplifying a line. If you wish to read/revise any theory at this stage, follow the links on the left.

Background Theory Links:

Digital Generalisation
Line Smoothing

Lang Simplification Algorithm Interactive Lessons:

Visualisation Exercise
Simple Interactive Example

Line Smoothing Algorithms

Cartographers produce manuscript lines which have a smooth ‘flowing’ appearance. In comparison, digital lines tend to be angular and non-aesthetically pleasing (particularly at large scales) – this is due mostly to the constraints of the digitising grid. Unlike simplification, which endeavours to reduce detail, smoothing techniques shift the position of points making up a line, in order to remove small perturbations and capture only the most significant trends of the line. Hence, smoothing is used to improve the appearance of digitised lines or, more simply, for cosmetic modification. Figure 3 shows a line before and after it has been smoothed.

Figure 3. Line Smoothing (Source: McMaster & Shea 1992).

As Figure 3 illustrates, the aesthetic appearance of a line can be greatly improved by applying one of the many smoothing operators. There are various ways that smoothing algorithms can be classified, see Table 1 below for one such classification.

Table 1. Characteristics of smoothing algorithms (Source: Lewis 1990 in McMaster & Shea 1992).

Category	**Description**
Point averaging routines	**a)** Calculates an average based on the positions of existing coordinate pairs and neighbours. **b)** Only the end points remain the same. **c)** Maintains the same number of points as the original line. **d)** Each algorithm easily adapted for different smoothing conditions by choosing different tolerance parameters. **e)** All algorithms are local or extended local processors.
Mathematical curve fitting routines	**a)** Develops a mathematical function or series of functions that describe the geometrical nature of the line. **b)** The number of retained points is variable and user-controlled. **c)** Retention of end points and of points on the original line is dependent on choice of algorithm and tolerances. **d)** Once the algorithm is chosen, there is little flexibility allowed in changing the final shape of the smoothed line. **e)** Function parameters can be stored and used to later regenerate the line at the required point density. **f)** There are local, extended local and global processing routines.
Tolerancing routines	**a)** Each algorithm uses some geometrical relationship between the points and a user defined tolerance to smooth the cartographic line. **b)** End points are retained, but the number of points generated for the

smoothed line and the number of interior points retained from the original data is algorithm-dependent.

c) Ability to change the curve's final appearance is algorithm-dependent.

d) There are local, extended local and global processing routines.

There are several line smoothing algorithms, including McMaster's Distance Weighting Algorithm, Boyle's Forward-Looking Smoothing Algorithm and Chaiken's Smoothing Algorithm. The algorithm at the focus of the line smoothing section of the module, though, is known as McMaster's Slide Averaging Algorithm. There is a link below (on the right) to one interactive lesson which will help you to visualise how McMaster's Slide Averaging Algorithm is applied when smoothing a line. If you wish to read/revise any theory at this stage, follow the links on the left.

Background Theory Links:	McMaster's Slide Averaging Algorithm Interactive Lesson:
Digital Generalisation Line Simplification	Visualisation Exercise

References

Hunter, G.J. 1998, 'Generalisation Techniques for Spatial Data', 451-620 Lecture Notes, The University of Melbourne.

McMaster, R.B. & Shea, S. 1992, Generalisation in Digital Cartography, The Association of American Geographers, Washington D.C., pp.3, 71-91.