Middlesex University London

Middlesex University Research Repository:

an open access repository of
Middlesex University research

http://eprints.mdx.ac.uk

Lupton, Kenneth, 1995.
Defining and using road network data in an accident database.
Available from Middlesex University's Research Repository.

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this thesis/research project are retained by the author and/or other copyright owners. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge. Any use of the thesis/research project for private study or research must be properly acknowledged with reference to the work's full bibliographic details.

This thesis/research project may not be reproduced in any format or medium, or extensive quotations taken from it, or its content changed in any way, without first obtaining permission in writing from the copyright holder(s).

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:
eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

DEFINING AND USING ROAD NETWORK DATA IN AN ACCIDENT DATABASE

Submitted by

KENNETH LUPTON

for the qualification of

MASTER OF PHILOSOPHY

Middlesex University
September 1995

Chiefs! Our road is not built to last a thousand years, yet in a sense it is. When a road is once built, it is a strange thing how it collects traffic, how every year as it goes on, more and more people are found to walk thereon, and others are raised up to repair and perpetuate it and keep it alive.

> *Valima Letters. Address to the Chiefs on the Opening of the Road of Gratitude, October 1894.*

ABSTRACT

This thesis proposes improvements to the design of road accident databases typically used by local authorities in England. The present design tends to lead to inconsistencies in the information relating to the road network contained in the database.

A methodology for the redesign of the database is proposed which will lead to greater data integrity and provide additional and more detailed information. The advantages of the system are demonstrated by producing accident predictive relationships for sharp bends and minor junctions.

The design has been carried out in the context of a relational database system incorporating data from a geographical information system. The advantages of an object-oriented system are also considered and proposed as a direction for further research.

ACKNOWLEDGEMENTS

The author would like to thank Professor Chris Wright and David Jarrett for their help in preparing this thesis, and also the local authorities who provided the data.

NOTATION

$S(x)$	spline function
$S_k(x)$	polynomial function
$S'_k(x)$	first derivative of polynomial function
$S''_k(x)$	second derivative of polynomial function
m_k	curvature of polynomial function
$B_{i,n}(x)$	B-spline
$E_w(f)$	sum of weighted squared residuals
$J(f)$	spline component representing energy due to bending
$K_w(f)$	minimising function for smoothing spline
T_i	tangent vector
V^i	first additional Bezier point
J_i	second additional Bezier point
W^i	third additional Bezier point
C_k	tension parameter
μ_i	mean of probability distribution
η_i	systematic part of a statistical model
g	link function
$E(Y)$	expected value of variable Y
D	scaled deviance
$l(b;y)$	log of the maximum likelihood
λ	mean of a Poisson distribution
χ^2	Pearson's chi-squared statistic
k	shape parameter of a prior gamma distribution
α	scale parameter of a prior gamma distribution
$\hat{m}$	mean of a posterior gamma distribution

LIST OF FIGURES

LIST OF TABLES

Contents

References

Appendix A

Appendix B

Appendix C

Chapter 1
Introduction

The accident databases currently used by local authorities in the UK are prone to error and provide incomplete information for the analysis of road accidents. This is partly due to the fact that the accident data is not related to an adequate definition of the road network. The purpose of this thesis is to compare the accident database systems currently in use and to develop a suitable road network definition using digital map data. It is also the intention to demonstrate the benefits of this system by developing accident predictive models using data not available from existing systems.

The overall aim of this work is to make a contribution to the efficiency of road accident database systems and to gain a more detailed understanding of the factors that influence accidents on the road network.

The data for this thesis was obtained during the 'Road Accident Migration' research project. This was an SERC funded project conducted jointly by Liverpool and Middlesex Universities. The purpose of the project was to establish whether the apparent migration of road accidents following an engineering improvement is a real effect or a statistical artefact [1,2,3]. Research work of this type relies heavily upon the data collected by local authorities during their work in road safety. The results of research provide an understanding of the phenomena of road accidents contributing to the common aim of their reduction. Road accidents are fortunately relatively rare events and can arise from a variety of circumstances. Investigations of this type require a large volume of accurate and reliable data in order to explore the relationships between the number of accidents and both the permanent and transient features of the road network.

During the migration project data was obtained from 13 local authorities. At Middlesex University data was provided by the Bedfordshire, Buckinghamshire, Durham, Humberside, Kent and Oxfordshire County Councils. Some data that was not subsequently used for migration analysis was provided by the Berkshire, Hertfordshire, Leicestershire and Suffolk County Councils. All of these counties did, however, provide a variety of accident database systems for comparison.

Of the authorities contacted the most detailed road network data was provided by the Kent County Council and it is this data which has mainly been used in this thesis. The network is shown below in figure 1. The roads are a selection of mainly rural major A and B roads in East Kent.

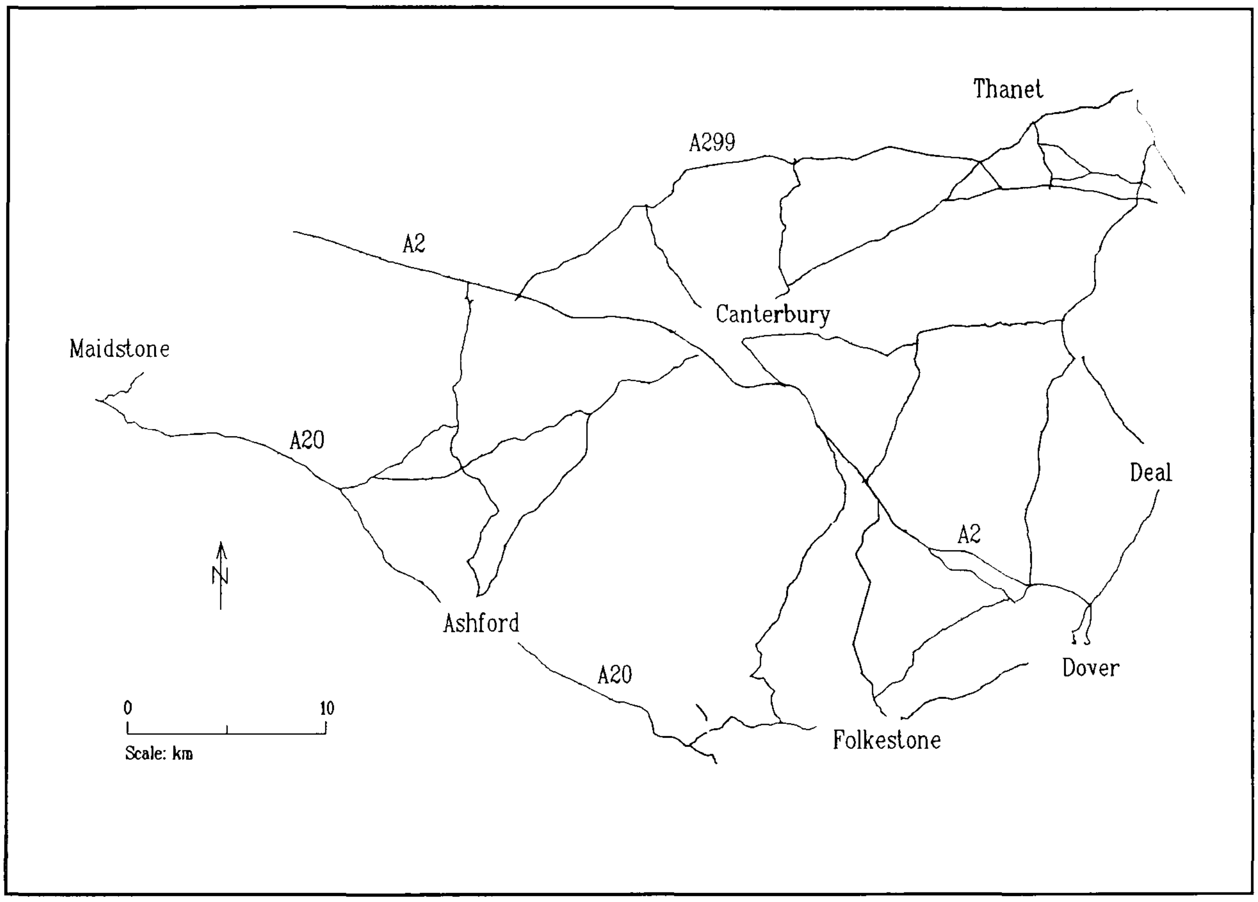


Figure 1.1 Study network in East Kent

Accident data, traffic flow data and highway improvement data were provided for the roads in Kent over an eight year period from 1984 to 1991. For the purposes of accident analysis parts of the network were not included between 1990 and 1991 where they were significantly affected by the construction of new roads, such as the M20 between Maidstone and Ashford and the dualling of the A299. These improvements caused either a redistribution of traffic or changed the character of the road in a way that were likely to significantly affect the number of accidents.

Chapter 2 describes various accident databases in use and the methods used for locating accidents on the road. It also describes the parameters currently used for categorising road types recommended by the Department of Transport that are incorporated in their computer program Coba 9. During the thesis some of these parameters will be examined with a view to improving them in the light of more detailed information about the road network from the database.

In chapter 3 fundamental aspects of the relationship between accidents and the road network are considered as the basis for the design of an accident database, assuming the use of a relational database system.

In defining the road network the first consideration is the method of representing the road centre line and this is considered in chapter 4. The possibility of extracting more precise geometry using cubic splines is considered, one advantage being that they can provide an estimate of centre line radius. In chapter 5, an application is demonstrated by examining the relationship between road accidents and curve radius.

After the centre line is defined other aspects of the network definition are considered in chapter 6, such as the method of representing junctions and bypasses. The potential for defining the road network in more detail using digitised road maps is illustrated by extracting accident data for minor junctions. This data is not easily available using current systems.

In conclusion, chapter 7 discusses the possibilities for using an object-oriented approach to road accident database design.

Chapter 2
Data Structures

2.1 Introduction

This chapter provides an introduction to database management systems, accident databases and geographical information systems and reviews the methods for locating accidents that were encountered on the accident migration project. It also describes some of the parameters commonly used to describe the road network in the Department of Transport cost benefit analysis program COBA9. Some of these parameters are considered in subsequent chapters with a view to improving them in the light of more detailed road geometry which can be made available from the database.

2.2 Database Management Systems

A database is a collection of non-redundant data which can be shared between different application systems. For example, departments within an organisation may need to view data from the database in different forms depending upon their requirements. Redundant data is data which is repeated unnecessarily in the database and can cause inconsistency in the data. A Database Management Systems (DBMS) is a generalised software interface which allows local views of the data in the database. DBMS have made an important contribution to the availability and coordination of data and their continuing development provides possibilities for the description of the road network in greater detail.

We relate to the road network via the road map which exists at various levels of detail to provide us with the information we require. The availability of maps in digital form facilitates the link between the map and the database which not only makes a large amount of data relating to a road feature immediately available on the computer screen but also enables the spatial relationships of the data to be investigated. These systems are known as Geographical Information Systems and at their core is the cartographic database.

The accident databases in use by local authorities incorporate the road accident data items recommended by the Department of Transport in the Stats 19 coding form. These items are listed in Appendix A. A variety of methods for locating road accidents were encountered during the research and the databases were generally designed before the relational database came into being and do not exploit its potential. The purpose of this thesis is to compare the systems in use and to explore their potential development in order to provide a more detailed and accurate description of the road network. This in turn provides a starting point for a detailed understanding of the relationship between accidents and features of the highway.

The efficient organisation of a local authority is partly dependent upon the ease of access to relevant and accurate data which enables investment decisions to be made most effectively. Increasingly over the past twenty years Local Authorities in the UK have been using computerised data base systems for storing information relating to the road network. DBMS are ideally suited to this purpose as they enable data to be available to regional offices and other departments from a central computerised system. Centralisation avoids the repetition of data where it is required by different offices which can cause inconsistencies when the data is updated. The greater accessibility of data means that a department can make planning decisions that incorporate information that is not within their direct control, for instance, an education department may require information about road accidents involving school children. In terms of the highway the applications of DBMS are varied:

Road Accidents.

The police provide a detailed description of each road accident that is reported to them. This information is used to establish common factors and trends in accidents and to identify accident blackspots.

Traffic flows.

Automatic traffic counters installed on the highway produce large volumes of accurate data. This data is essential for monitoring trends in traffic growth which in turn enables the implications of alternative regional planning policies to be assessed more effectively. The likely effect and cost benefits of new road schemes can also be more accurately predicted.

Maintenance.

A DBMS can be used to store information related to the condition and maintenance needs of the road network and enable investment decisions to be made with consideration to the needs of the entire network.

The rapid development of database technology means that existing systems can rapidly become out of date and although the new systems may offer considerable advantages the change from one to another may be a difficult and expensive process. An organisation may be reluctant to undertake a change unless they are quite sure of the benefits. The purpose of this thesis is to consider the methods of organising road accident and the benefits that can be obtained.

2.3 Road Accident Data

The Department of Transport collects road accident statistics from the police authorities across the country to detect and analyse national trends. The information required is contained in the STATS 19 coding form, see Appendix A. This information is usually incorporated into the local authority accident database which often contain additional attributes.

An accident database typically consists of three tables:

1. Attendant circumstances.
2. Casualty table.
3. Vehicle table.

Attendant circumstances.

This table contains information relating to the date and location of the accident including the road number and grid coordinates. Data relating to the road environment is included such as road and junction type, speed limit and street lighting conditions. It also contains information relating to particular circumstances surrounding the accident such as the weather conditions and the condition of the road surface.

Casualty table.

This table contains information about the casualties such as the severity of their injuries, their age and sex and whether they were drivers, passengers or pedestrians. If a casualty is a schoolchild then the name of the school is included.

Vehicle table.

This table includes a description of the vehicles involved such as their type, age and size and their behaviour during the accident. Information pertaining to the driver is included, such as the result of any breath test.

All of this information is usually collected by a police officer present at the scene of the accident.

2.4 Location of Accidents

An important piece of information about an accident which is essential for analysis is its location. The police officer who records the accident locates it by providing a distance to a road feature such as a junction and also by its grid coordinates. The grid coordinates are often not precise, for example, the nearest 100 metres, and local authorities often implement checking procedures to validate the position. During the accident migration project four systems were encountered for locating accidents:

1. Coordinate system.
2. Chainage system.
3. Bobstar sections.
4. Link and node system.

Coordinate system.

In this system accidents are located by the road number and Ordnance Survey grid coordinates alone. Manual checking procedures are usually implemented by the local authority to ensure the accuracy of the coordinates. The disadvantage of this system is that no spatial relationship between accidents is available.

Chainage system.

The location of a point on the road is measured usually from Ordnance Survey plans. The start point of the road is usually a junction or local authority boundary. Accidents are easier to locate using this method since the police usually reference them by a distance from a road feature, in addition to their grid coordinates.

Accidents within 20 metres of a junction are usually given a single chainage. In the case of roundabouts an allowance for the distance around junction may be included if the system is also used for road maintenance records.

Bobstar sections.

Bobstar is a complete accident database system developed by the Berkshire, Oxfordshire and Buckinghamshire County Councils in the 1970s.

In this system the road is divided into sections that are often approximately 250 metres long in rural areas but this varies from one authority to another. Each section is intended to contain details of accidents related to the same feature of the road. In some cases the length of a section is approximately inversely proportional to the number of accidents that are likely to occur on that section.

In urban areas the length is usually fixed at 100 metres. Major junctions are also allocated a section. The start of the section is taken to be 40 metres from the give-way line but again this varies depending upon the junction layout. Sections also start and end at changes in carriageway type, sharp bends, speed limits and local authority boundaries.

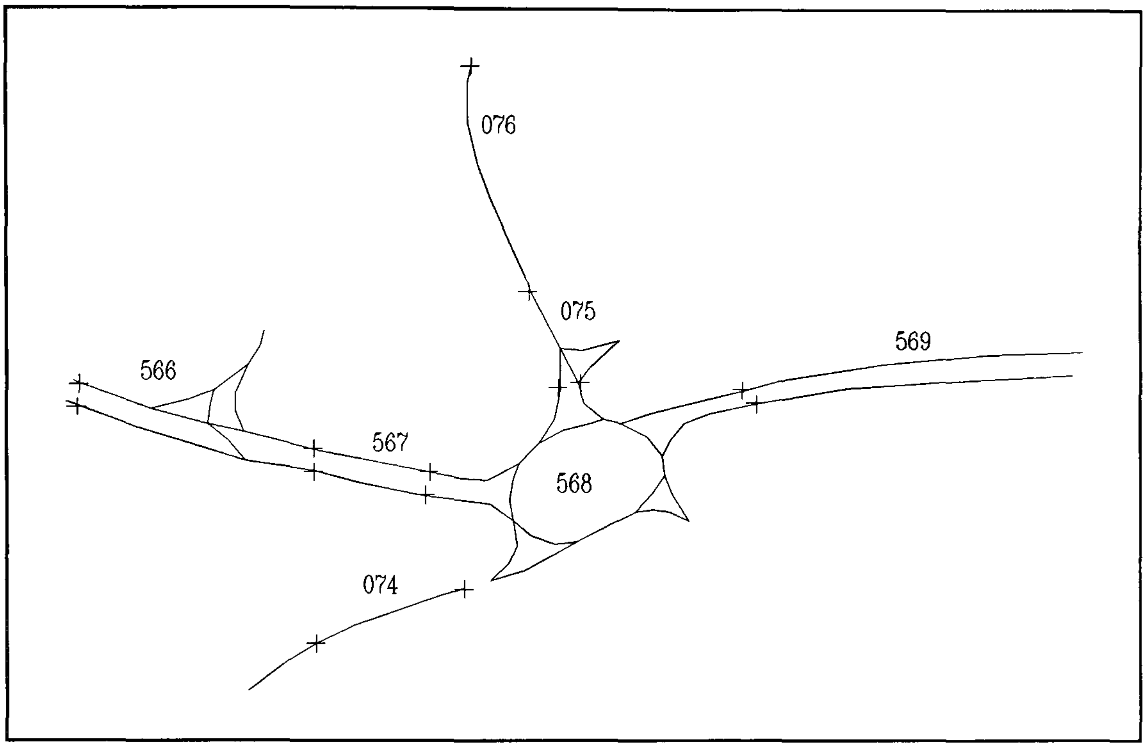


Figure 2.1 Bobstar Sections

Accidents are located by their road number, section number and grid coordinates. The advantage of this system is that it enables accidents relating to a specific road type or feature to be easily identified and is not dependent upon a high degree of locational accuracy. The disadvantage is that in rural areas it incorporates no measure of distance since the length of the sections varies and is generally it is not known precisely.

Link and node system

In the link and node system a node is defined as a major junction and has a unique reference number and is located by its grid coordinates. A link is the section of road between two nodes and it has its own chainage system and reference number. Slip roads are referenced to a particular junction and numbered individually.

The link and node system incorporates the advantage of the chainage system in that accidents are easier to locate manually and the chainage provides a spatial relation between them. In the chainage system accidents are allocated a road number and chainage and this means that the accidents for a particular road form a large unit of data which makes search operations inefficient. Dividing the road into links reduces the size of the data blocks which in turn speeds up search operations.

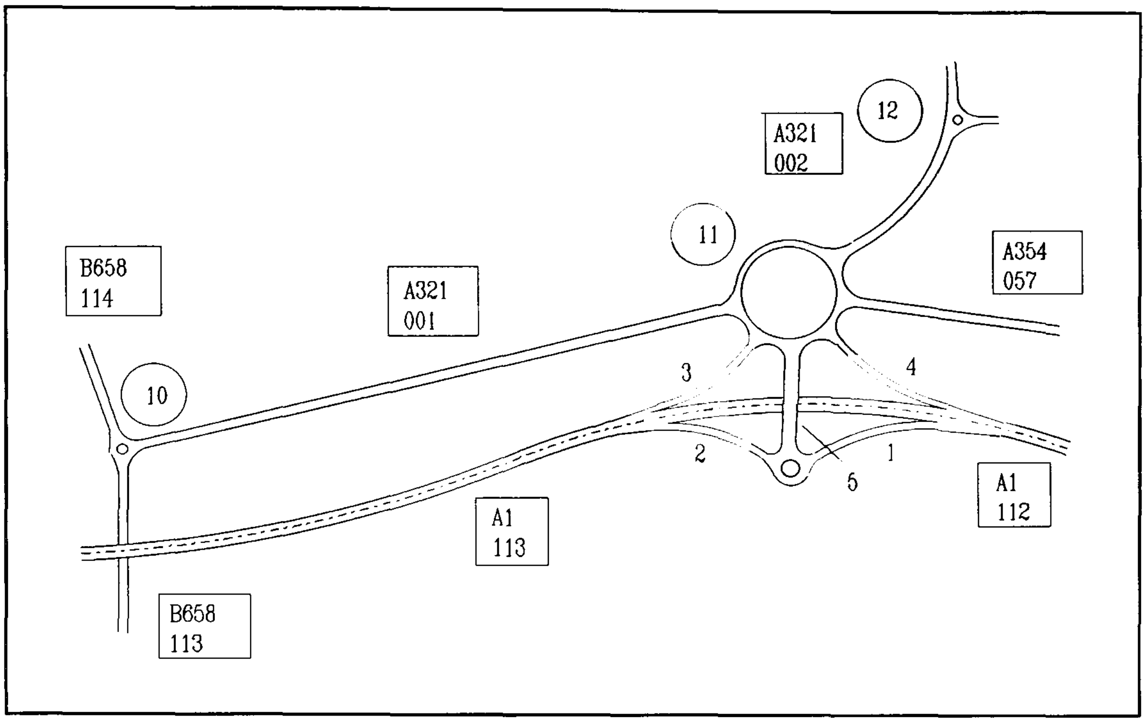


Figure 2.2 Link and node system.

The disadvantage of all four of these systems for locating accidents is that the definition of the road network is not contained within the database and separate reference must be made to a map. Elements of the network only exist, in effect, after an accident has occurred so it would not be possible to ask the database a question such as "Which major roundabouts had no accidents in 1993?" If one purpose of the database is to identify sites with more than a fixed number of accidents then this may not be a problem; however, if the risk at a site is to be measured by comparison to other similar sites then it is a serious omission. If a lightly trafficked junction regularly experiences two accidents a year then it may not qualify as a blackspot but if other similar junctions experience none then it is worthy of investigation.

2.5 Geographical Information Systems (GIS) [4,5,6]

A geographical information system is designed to store, manage, display and analyse all types of geographic and spatially related data. A GIS combines a Database Management System with the ability to represent data in the form of maps and other graphic displays. It has the ability to combine different data sets, for example census information, land use data and highway network data, to form a database which enables the modelling of many forms of geographic phenomena.

In a GIS a map is made up of a series of layers each representing a set of geographic features; for instance, one layer may display rivers, a second the ground contours and a third land use. The thematic and locational attributes for each layer is held in a table which is part of the cartographic database.

The locational attributes may be stored by three different methods, points, lines and polygons, see figure 2.3. Road accidents would be shown as points, roads and rivers represented by lines and postal districts or land use represented by polygons. A layer may contain a mixture of these types for instance spot levels would be represented by points but contours by polygons. A GIS usually has the facility for calculating the perimeters and areas of polygons.

For the road network a GIS can be used to store information relating to the location of the roads, type of road, maintenance history, traffic flow and accident data. It is possible to display and analyse entities possessing particular attributes recorded in the database, for example, the number of accidents occurring in the hours of darkness. Since the location of entities is defined, it is possible to identify features within a certain geographical area, for example, it is possible to calculate the distance between a school and the accidents involving pupils of that school.

Advances in technology have made it possible to encode large volumes of digital map data. The location of geographical features can be determined using satellites, known as Global Positioning Systems. Remote sensing and photogrammetry using aeroplanes and satellites are possible sources of data as are existing maps that can be digitised or scanned. These advances combined with improvements in the processing speed and storage capabilities of computers means that a GIS is now a powerful tool for the analysis of traffic related problems.

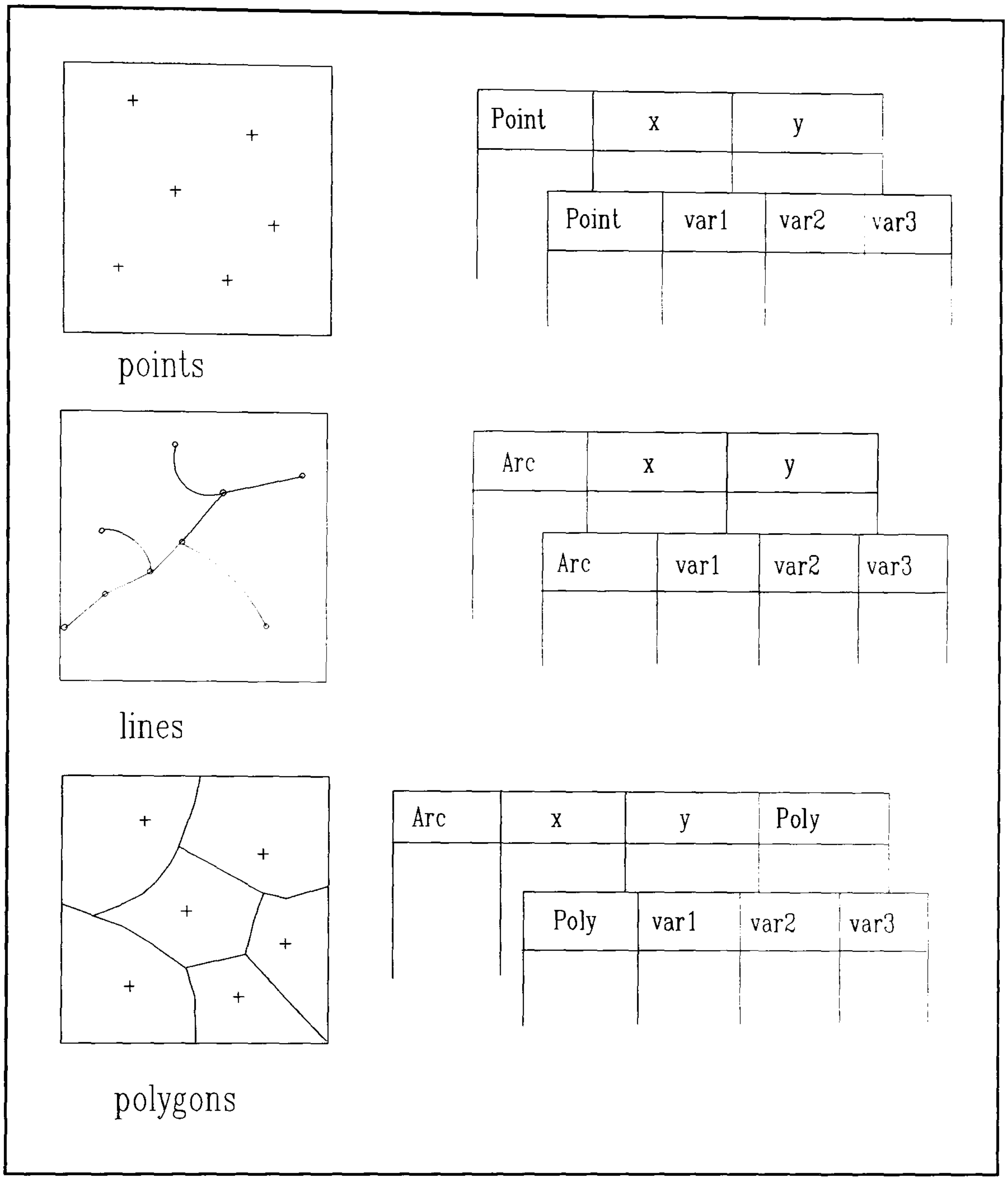


Figure 2.3 Cartographic database.

In practice local authority GIS are developing independently using different digitised maps and software. Ideally all would use a single national system controlled centrally, with each local authority department having access to their relevant data. This would enable accident statistics to be collected at a national level and local authorities to carry out more detailed analysis within the same system. The Ordnance Survey is now producing a digitised road network, OSCAR, which covers every public road in the country. It is currently available at three levels of detail:

Network Manager. A simplified representation of the major road for network and route planning.

Route Manager. This represents the adopted road network in a more detailed form but with junctions represented as single nodes.

Asset Manager. Contains detailed large scale road geometry and it includes all public roads and some private roads. It uses a link referencing system with links identified between each junction and intersection including minor roads. Individual elements of more complex road configurations such as grade separated junctions or roundabouts are also referenced individually.

2.6 Coba 9

The Department of Transport have developed methods for classifying road types for: the calculation of speed flow relationships, travel time costs and the prediction of accidents. These methods are explained in detail in the Department's cost benefit analysis program, Coba 9, reference manual [7].

Coba 9 is a cost benefit analysis program for the economic evaluation of new road schemes. It provides a monetary evaluation of the benefits accrued to the road user over thirty years as savings in:

i) Accidents.
ii) Time.
iii) Vehicle operating Costs.

These elements are combined with:

i) Capital cost of the schemes.
ii) Road maintenance costs.

Coba evaluates the costs of the existing network to provide a basis for comparison for the costs of alternative new schemes.

ACCIDENTS

Coba requires the user to define the road network as a system of links and nodes.

Links are classified in terms of road types and nodes, and where appropriate as junction types. Once the traffic flows for these elements are calculated then the accident rate, expressed in the number of personal injury accidents per million vehicle kilometres, can be estimated from equations developed for Coba. Existing roads are described initially as being rural, speed limit > 40 mph, or urban, speed limit ≤ 40 mph. The types are:

- Rural
    - A
    - B
    - Other
    - Motorway

- Urban
    - A
    - B
    - Other

Where a link contains a large number of junctions a combined accident rate is given for each category. Junctions are classified according to whether they are rural or urban, their type and their number of arms. The types are:

- Major/minor
- Traffic signalled
- Standard Roundabouts
- Small Roundabouts
- Mini Roundabouts

TIME SAVINGS

Time savings are the major source of cost benefit from a new road. A relationship is required between the traffic flow and vehicle speeds for a given road. Nationally derived relationships for different classes of road are used in Coba. The classes are:

1. Rural single carriageway, nominal width 7.3m.
2. Rural single carriageway, nominal width 10m.

3. Rural all purpose dual carriageway.
4. Motorway dual two lanes.
5. Motorway dual three lanes or more.
6. Suburban main road, 40mph speed limit.
7. Urban non-central, 30 mph speed limit.
8. Urban central.

9-12. Special user defined relationships.

The attributes required to determine the speed flow relationships for the category of rural single carriageway roads are as follows:

1. Bendiness; total change of direction per unit distance (deg/km).
2. Hilliness: total rise and fall per unit distance. (m/km).
3. Net Gradient: net rise per unit distance (m/km), one-way links only.
4. Vehicle flow per standard lane (veh/hr).
5. Average carriageway width (m).
6. Verge width.
7. Total number, both sides, of laybys, side roads and accesses, excluding house and field entrances. (no/km).
8. Average sight distance (harmonic mean) (m).

Similar attributes are required for the other road types.

It is apparent that the road network is defined in greater detail for the determination of speed flow relationships than the prediction of accidents. Many of the geometric variables in Coba are intended to be extracted manually from a plan. The availability of digitised road maps means that they can now be calculated automatically and in some cases there is scope for defining network elements in more detail. In the ensuing chapters some of these possibilities are explored. In the next chapter, however, the fundamental concepts which govern the design of an accident database are considered.

Chapter 3
Accident Database Design

3.1 Introduction

In this chapter the design principles of the accident database are reconsidered assuming the use of a relational database system. The intention is that this should lead to a more efficient system and one which is less prone to data error. The Department of Transport require information about road accidents to obtain detailed statistics at a national level. The information required is listed in the Stats 19 coding form and is shown in Appendix A. A local authority database usually incorporates this data plus any local information that they require. It is this data which forms the basis for the database design considerations in this chapter. First it is necessary to describe the principles of relational database design.

3.2 The Relational Database [8]

The relational database system is the most generally used system at the present time. Compared to its predecessors it is more efficient in terms of storage, and is more versatile since it allows data to be extracted and manipulated by the user in many forms depending upon requirements. In a local authority different departments may require information from the same data set. This can be achieved using applications written by the user that are incorporated into the database system. These applications are usually written in the *Structured Query Language*, SQL, a programming language developed specifically for relational databases.

In a relational database the data is divided into subsets called *tables*. A table is made up of columns headed by the name of an *attribute type* and rows, often known as *records*. The intersection of a row and column contains an *attribute value*. The attribute values that make up a particular row are not interchangeable with those in other rows since the system sets up relationships between them. Columns can be added or removed from the table without compromising its validity, but more important, columns can be added from other tables provided a relationship is established between the tables. This relationship can be made by including a *primary key* which is a column, or columns, common to both tables that uniquely defines each row.

3.3 Table Design

The data in a table should be organised so that it is *normalised* and that it does not contain *redundant* data. There are three conditions for the normalisation of a table:

1. Each row must be distinct and not duplicated.
2. Attribute values which are not primary keys must be fully dependent upon the primary key.
3. Every non-primary key must be non-transitively dependent upon the primary key.

Redundant data is data which can be eliminated from a table without information being lost. *Duplicated data* occurs when an attribute has two or more identical values. These principles are illustrated in Table 3.1 where three separate accidents are recorded at the same junction and the junction number and junction type are duplicated. The junction number is not redundant since it provides the location of each accident, however the junction type is redundant since it can be found from the junction number. In this table the junction type is transitively dependent upon the primary key, the accident reference number, and violates rule 3 above. The redundancy can be eliminated by splitting this table into two normalised tables, accident location in table 3.2a and junction type in table 3.2b. A relational join can be established between these two tables via the attribute type junction number.

Acc_Ref	Jnc_No	Jnc_Type
A0001	J001	RDBT
A0002	J001	RDBT
A0003	J001	RDBT

Table 3.1 Redundant and duplicated data.

Acc_Ref	Jnc_No
A0001	J001
A0002	J001
A0003	J001

Table 3.2a. Accident location

Jnc_No	Jnc_Type
J001	RDBT

Table 3.2b. Junction type

Acc_Ref	Chainage	Sp_limit	St_Lights	Jnc_Type
A0123	1.39	60	-	Y
A0124	1.39	40	+7m	Y
A0125	1.39	40	+7m	T
A0126	1.39	60	+7m	T
A0127	1.39	40	-7m	T
A0128	1.39	40	-7m	T

Table 3.3 Data errors associated with redundancy.

The unnecessary repetition of data is inefficient in terms of the acquisition and input of data and it is also prone to error. Table 3.3 shows six accidents at the same location at different times. Due to the errors in this table it would be difficult to determine the junction type, speed limit or the height of the street lighting at chainage 1.39 with any degree of confidence.

An alternative explanation for the inconsistency in Table 3.3 is that the junction type, speed limit and street lighting have been changed during the study period. While this information is not usually included in an accident database it does illustrate the risks in drawing conclusions after simply inspecting a subset of the data without establishing a structure which governs the type of data to be included in the database.

3.4 Entity relationships.

The relationship between the entity accident and {location,time} is an example of a "one to one" relationship. The relationships between location and accident, and time and accident are both "one to many" relationships since many accidents can occur at the same location at different times and many accidents can occur simultaneously at different locations. Other examples of "one to many" relationships are those between accident and casualties, and accident and vehicle since there may be more than one casualty or vehicle involved in an accident. Each of these is referenced to the accident in which they were involved. One piece of information which is not available is whether the individual person or vehicle has been involved in more than one accident. This could only be determined by linking the database to census data or vehicle registration data and this may be considered impracticable.

On the road network, a location can be defined as a combination of road number, and either chainage or grid coordinates, and time by the date, hour and minute of the day. However, this is not necessarily unique as shown in the following situation:

> A driver looks across the road at the vehicles travelling in the opposite direction and sees an accident developing, accident A. He is so distracted by this that he drives into the car in front, accident B. Should this be reported as one or two separate accidents?

Since both accidents occur at the same place and time perhaps they should be reported as a single accident but intuitively one might say that they are separate. One solution might be to add the lane number to the definition of location, but this could cause problems in the more common situation where an accident involves vehicles in more than one lane. As an illustration, the following example was taken from an accident database. Two accidents were reported at the same time by two different police officers on a dual carriageway, one on the northbound carriageway, the other on the southbound carriageway and the verbal descriptions of the accidents are as follows:

1. A1-N. Car towing trailer, vehicle 1, travelling north, struck vehicle 2 also travelling north and crossed the central reservation and struck vehicle 3 travelling south.

2. A1-S. Car travelling south, vehicle 1, was struck by car towing trailer travelling north, vehicle 2, which had crossed the central reservation after colliding with vehicle 3 also travelling north.

In this database the subdivision of the road into two separate carriageway has contributed to the double counting of the same accident.

Returning to the original problem, there is another apparent anomaly if one considers the vehicles involved. Vehicles are included in the database that contribute to the accident and so the vehicles of accident A have contributed to accident B but not vice versa. Including vehicles A in accident B would break the rule that the same vehicle cannot be involved in more than one accident which would lead to the double counting of vehicles. The solution to this problem is offered in the attribute type *carriageway hazard,* see Appendix A, which can take on the value *previous accident* and is applicable to accident B.

Another situation is one where a vehicle collides with a stationary vehicle that has been involved in a previous accident. Whether this is reported as one or two accidents usually depends upon the time difference involved. If the second collision occurred seconds after the first collision then it would be reported as one, but if the difference was, say, thirty minutes it would be reported as two. If a circumstance occurs on the road which causes an accident there is an associated probability that a certain number of vehicles will be involved, and also a probability that they will become involved after a certain time interval. To preserve the concept of statistical independence therefore it should be thought of as one accident. However this would lead to a loss of information, possibly relating to the effectiveness of precautions taken at the scene of an accident to warn oncoming drivers. Again the attribute value *previous accident* is appropriate, but perhaps these accidents should not be included in any statistical analysis which assumes independence.

An additional requirement of the database can therefore be made that any accident which occurs at a different time or place and may be considered separate to any other accident should be recorded as a unique event.

3.5 Elimination of Redundancy

Recently, Austin [9] has suggested a validation procedure by comparing the recorded locational variables with the same variables defined separately in a GIS. The variables identified were: road class, road number, district, speed limit, pedestrian crossings, junction control, junction detail, carriageway type and markings.

In the attendant circumstances table the non-redundant data is that which is relevant to the accident at the time that it occurred; the redundant data is that which is descriptive of permanent features of the road network unless it describes the location of the accident. The redundant data is listed as follows:

1. *link number*
2. *speed limit*
3. *road type*
4. *junction detail*
5. *junction method of control*
6. *lighting conditions*

Within these attribute types however, there are categories that are not permanent features or which may not be defined as part of the network:

> *Junction detail.* This includes private entrances. It is assumed that the roads defined in the network are those within the jurisdiction of the local authority.

> *Junction method of control.* This includes control of the junction by an authorised person who could be included as a separate column.
>
> *Lighting conditions.* While the type of lighting is a permanent feature provision must be made to show whether the lighting was on or off at the time of the accident, again this could be recorded separately.

One attribute type that is not included in this list is that of *pedestrian crossing* and this requires more detailed consideration, see attendant circumstances in Appendix A. The categories of crossings are very detailed and it is not appropriate to define all of them in separate tables, for example, it would be impracticable to define every central refuge on the network. However, there is a source of redundancy since there is repetition in the type *pedestrian location* in the casualty table. The reason for this is that a pedestrian may be a contributory factor in an accident without necessarily becoming a casualty. This redundancy can be eliminated by defining the type and location of crossing in a separate table and eliminating central refuge from the attendant circumstances. It has already been mentioned that the presence of an authorised person can be defined separately.

Some authorities include the attribute type *overtaking manoeuvre* which describes the number of vehicles travelling in each direction and the number overtaking. This presumably applies to moving vehicles only. This provides another source of redundancy compared with the attribute type *manoeuvre* in the vehicle table which gives a detailed description in three main categories; overtaking or colliding with stationary vehicles, other moving vehicles or vehicles manoeuvering at junctions. If *overtaking manoeuvre* is intended to apply to vehicles overtaking at speed only then in practice this information is often inaccurate since it tends to be recorded for any overtaking manoeuvre.

Some attribute types in the attendant circumstances table that can be determined from other tables are *severity, number of casualties* and *number of vehicles*. Many SQL implementations incorporate a function that will calculate the day of the week from a specified date.

3.6 Enterprise Rules

In order to develop a conceptual model of the database and to eliminate redundant data, it is useful to determine the *enterprise rules* that establish the relationships between different data types, for example that:

1. A casualty is associated with one accident.
2. An accident may involve several casualties.
3. At least one vehicle is associated with one accident.
4. An accident may involve several vehicles.
5. There is one {location,time} associated with each accident.
6. There is one road type associated with each {location,time}.
7. There is one junction type, where applicable, associated with each {location,time}.
8. There is one junction control method associated with each junction.
9. There is one speed limit associated with each {location,time}.
10. There is one type of street lighting associated with each {location,time}.

The database now could be modified to contain the following information in separate tables:

Accident detail.

1. Attendant circumstances
2. Casualty
3. Vehicle

Network detail

4. Road type
5. Speed limit
6. Junction type

7. Street lighting.
8. Pedestrian crossing.

This structure has three main advantages over the current databases which are:

1. The tables are normalised.
2. The network data can be stored in a GIS.
3. Additional network attributes may be added without requiring modifications to the attendant circumstances table.

The development of the network tables will be discussed further in chapter 6.

3.7 Conclusion

The design of the Stats 19 accident database, currently in use by local authorities, was conceived before the advent of the relational database. It contains redundant data which can lead to data errors. The number of errors and the volume of data to be recorded can be minimised by redesigning the database into normalised tables. It has been necessary to develop rules and definitions that govern the data to provide a framework for the database. A major source of redundancy in the present system is that the road network is not defined in separate tables.

The first consideration in defining the road network is the method of representing the road centre line. This problem is considered using data from digitised road maps in chapter 4.

Chapter 4
Highway Geometry

4.1 Introduction

There are a number of circumstances where the highway engineer needs to know the geometry of an existing road. When designing the tie-in for a new road the engineer needs to know the precise location of the existing centre line and also its radius to provide a smooth transition. Knowing the centre line radius would also enable the relationship between radius and accidents to be investigated and also the deterioration of skid resistance with radius.

The traditional method of finding the radius of an existing road is to overlay templates of known radii, such as railway curves, on a large scale plan until a satisfactory match is obtained. Where more precise information is required it is obtained from a site survey.

In the case of a recently constructed road this information may be available from the construction records, but for the majority of roads this does not apply. Many roads have been in existence for centuries and their present alignments are the result of a succession of improvements. Whilst a new road may be designed as a series of straight lines and circular arcs or cubic splines, the alignment of an existing road may be more complex.

In this chapter methods are considered for obtaining geometric information about the road alignment from the data stored within a GIS.

4.2 Centre line data

A digitised road alignment provides new source of information about the geometry of a road. It was decided to explore the possibility of using this data to represent the road more fully. The computation of road geometry was carried out using data supplied by the Kent County Council. The data comprised road alignments that included centre line and junction detail extracted from their GIS. The data were supplied on disk in the standard drawing interchange format, *dxf*. A *dxf* file is an ASCII file which list the entities and their locations that make up a drawing. These files were viewed and edited using Autocad since it is a powerful graphics package readily available in the University and is able to import and export *dxf* files.

It was found that the supplied centre lines were made up of a series of short arcs that were not always connected and also were not necessarily in order. Before any centre lines could be defined the arcs had to be sorted into road and chainage order. This was achieved by redrawing the centre lines as single entities using Autocad. First extraneous information such as junction detail and slip roads was removed to leave only the arcs that made up the centre line. The details of these arcs were then stored in a *dxf* file created by Autocad.

A program was written which extracted the coordinates of the end points of the arcs that made up the centre lines from the *dxf* file. The program also produced an Autocad script file which contained commands for Autocad to replot the coordinates as points. Once the points were plotted a series of connecting lines, or polylines, were then drawn through the points to produce a single centre line for each road. The coordinates of each centre line, that were now in order, were then extracted once again from a *dxf* file and stored in the relational database system, Foxpro2.

4.3 Road length

A fundamental description of a road is its length. In order to define a chainage system an SQL program was written which calculated the cumulative distance along the road. This program CHAIN.PRG is listed in Appendix C. Part of the resulting output is shown below in Table 4.1.

Road	Easting	Northing	Chainage
A0251	601444	160467	18.487
A0251	601451	160520	18.540
A0251	601459	160570	18.591
A0252	594928	149715	0.000
A0252	595000	149732	0.074
A0252	595164	149771	0.243

Table 4.1. Road alignment and chainage.

4.4 Accident Location

The accidents on these roads were referenced to this chainage system. Accidents are located by their grid coordinates, that generally are close to but may not lie on the road centre line and so it was necessary to write a program which located the perpendicular offset from the accident to the centre line in order to calculate the chainage. This is calculated by vector geometry, and the method is shown in detail in Appendix B and summarised here.

First the distance between the start of a centre line arc and an accident is calculated and also the angle between this line and the link. This enables the calculation of the intersection point between the perpendicular offset from the accident and the arc.

Initially a hypothetical road with accidents was used to develop the program, see figure 4.1. In figure 4.1 it is clear which section of road each accident is related to, however some accidents have perpendicular offsets to more than one link, for example *a8* has offsets to three links. In order to select the correct link a zone, nominally 20 metres wide, was defined either side of the centre line and only accidents that fell within this zone were allocated a chainage on that link. This also provides a check on the accident location since any accident erroneously placed outside of this boundary might well have a lateral error also. Accidents not within the boundary appear in the output with a chainage of zero. They were then relocated to their correct position according to the accident record.

There is an additional problem in the vicinity of nodes, see figure 4.2. Accident *a*1 is within the boundary and has an offset to both links, whereas *a*2 does not have an offset to either link. The solution was to locate accidents within a 20 metre radius of a node and to give those the chainage of the node. The program ACCLOC.PRG is listed in Appendix C and a sample of the output in table 4.2.

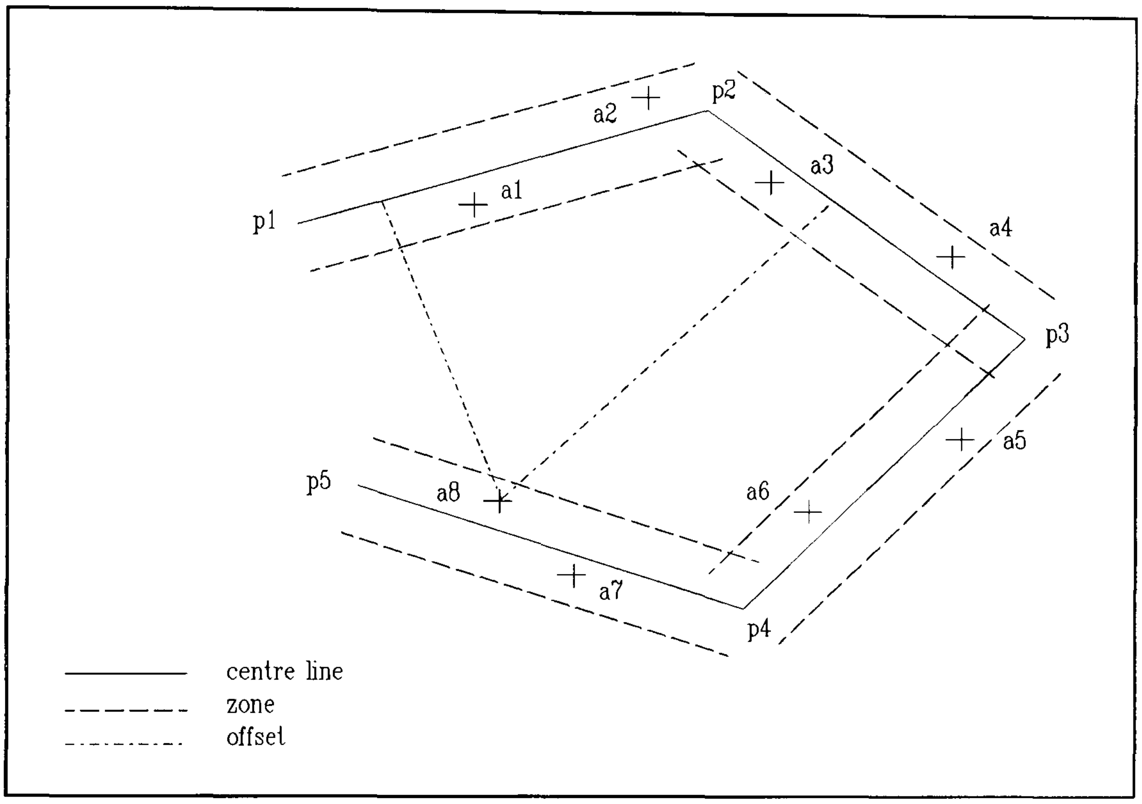


Figure 4.1 Hypothetical road and accidents.

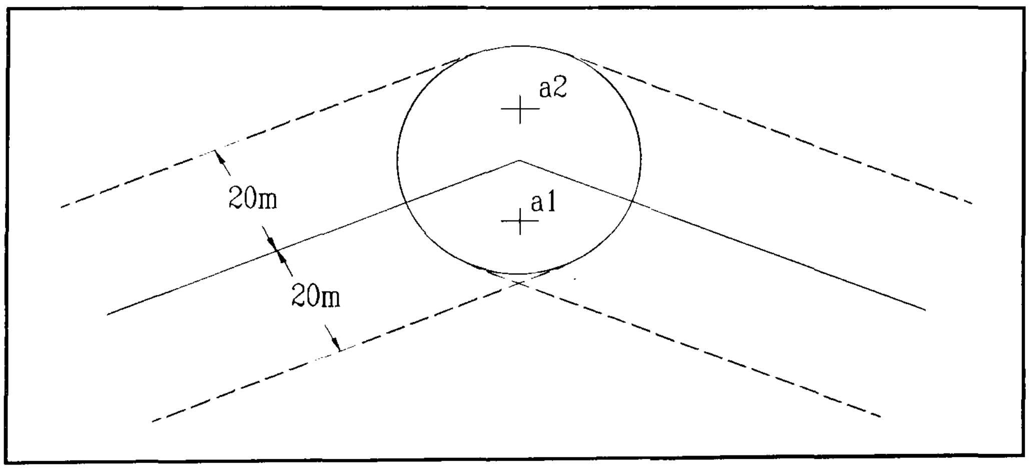


Figure 4.2 Accidents near a node.

ACCID	ROAD	EASTING	NORTHING	CHAINAGE
A3045	A0257	620731	157528	5.70
A3046	A0257	620742	157523	5.71
A3047	A0257	620756	157516	5.72
A3048	A0257	620822	157477	5.81
A3049	A0257	620633	157470	5.82
A3040	A0257	620903	157422	5.90

Table 4.2 Output from Accloc.prg calculating accident chainage.

4.5 The representation of digitised horizontal alignments

In order to calculate speed-flow relationships COBA incorporates the variable *bendiness* in order to represent the variability of the horizontal alignment. Bendiness is defined as the total change in direction per unit distance (deg/k). This variable can be readily calculated knowing the deflection angle between digitised centre line arcs. The deflection is estimated as follows:

Consider three successive data points p_1 (x_1,y_1), p_2 (x_2,y_2), p_3 (x_3,y_3). (Figure 4.3).

Let $a = x_2 - x_1$, $b = y_2 - y_1$,

$c = x_3 - x_2$, $d = y_3 - y_2$.

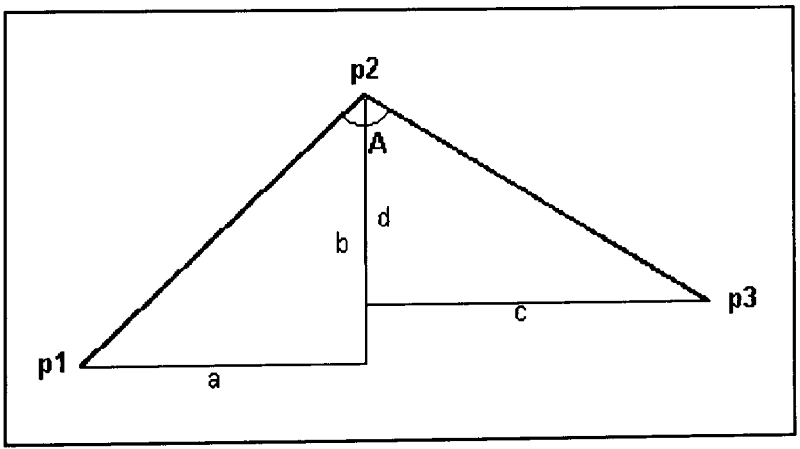


Figure 4.3 The included angle A between two lines, p_1p_2 and p_2p_3

The distance l_1 from p_1 to $p_2 = \sqrt{(a^2 + b^2)}$ and

the distance l_2 from p_2 to $p_3 = \sqrt{(c^2 + d^2)}$
the direction cosines for p_1 to p_2 a/l_1, b/l_1
the direction cosines for p_2 to p_3 c/l_2, d/l_2

The angle A between the two lines is given by the expression

$$\cos A = \frac{ac+bd}{l_1 l_2}$$

This calculation has been implemented in the Foxpro2 program ANGMES.PRG, see Appendix C.

The availability of digitised road alignments allow a more detailed representation of the alignment. In the Bobstar system sharp bends are allocated a section, however the radius is not included. Highway engineers have used cubic splines for the design of horizontal alignments for a number of years and an investigation was carried out to assess the feasibility of fitting splines to a digitised alignment. When designing a road the designer fixes the curve at a few pinch points and allows the spline to develop a natural shape between. For a digitised alignment, however, the data points are far more numerous and close together and so the spline has more constraints. The investigation was carried out using data made available by the Kent County Council and the Oxfordshire County Council whose alignments were digitised from the 1/1250 and 1/2500 Ordnance Survey.

4.6 Cubic Splines [10].

Theoretical Considerations.

It is possible to construct a single polynomial which passes through any given set of data points, however, functions of this type can produce undesirable oscillations. A more reliable method is to piece together a series of lower order polynomials, or splines, that pass through successive points. The resulting curve is therefore made up of a series of shorter overlapping curves or splines. There are two requirements for this method, firstly that the curve should fit the data points with a smooth alignment and secondly that a sufficient number of conditions should be specified to enable the calculation of the polynomials.

If the curve is to have a smooth appearance then at the point where two splines meet, known as a *knot*, they should have the same radius of curvature. This implies that the curvature should be continuous. The second derivative of the polynomial should therefore be at least a linear function. The minimum order of polynomial that satisfies this condition is the cubic.

Consider the $n+1$ points (x_k, y_k) where $x_0 < x_1 < \ldots < x_n$ The function $S(x)$ is called a cubic spline if there exist n cubic polynomials $S_k(x)$ such that:

1. $S(x) = S_k(x) = y_k + s_{k1}(x-x_k) + s_{k2}(x-x_k)^2 + s_{k3}(x-x_k)^3$
2. When $x = x_k$ then $S_k(x) = y_k$.

   The spline passes through each data point.
3. $S_k(x) = S_{k+1}(x)$.

   The spline forms a continuous function.
4. $S'_k(x) = S'_{k+1}(x)$.

   If the first derivatives are equal then the slope is continuous.
5. $S''_k(x) = S''_{k+1}(x)$.

   If the second derivatives are equal then the curvature is continuous.

Two additional conditions, the *endpoint constraints,* enable the calculation of the spline. If the endpoint constraints are that the curvature is zero then this is the condition for a natural cubic spline. Since most road alignments start and end at a junction where the curvature of the centre line less important than the junction layout this is the condition that has been used.

The curvature $S''(x)$ when plotted against x forms a set of linear splines, where $S''_k(x)$ exists if $x_k <= x <= x_{k+1}$ and $k = 0,1,\ldots\ n\text{-}1$. Any point on this system of straight lines can be calculated using the Lagrange interpolation formula,

$$S''_k(x) = S''_k(x_k)\frac{x-x_{k+1}}{x_k-x_{k+1}} + S''_k(x_{k+1})\frac{x-x_k}{x_{k+1}-x_k}$$

Integrating this equation and incorporating the above conditions yields the following cubic expression,

$$S_k(x) = \frac{m_k}{6h_k}(x_{k+1}-x)^3 + \frac{m_{k+1}}{6h_k}(x-x_k)^3 + (\frac{y_k}{h_k}-\frac{m_k h_k}{6})(x_{k+1}-x) + (\frac{y_{k+1}}{h_k}-\frac{m_{k+1} h_k}{6})(x-x_k)$$

where $\{m_k\}$ represents the curvature which can be calculated from the expression

$$h_{k-1}m_{k-1} + 2(h_{k-1}+h_k)m_k + h_k m_{k+1} = u_k$$

where $u_k = 6(d_k - d_{k-1})$ for $k = 1$ to n-1 and $d = (y_{k+1}-y_k)/h_k$ and $h_k = x_{k+1}-x_k$ for $k = 1$ to n-1.

These equations together with the endpoint constraints yield the following system of linear equations

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & 0 & .. & 0 \\ h_1 & 2(h_1+h_2) & h_2 & .. & 0 \\ 0 & .. & .. & .. & .. \\ .. & .. & .. & .. & h_{n-2} \\ 0 & .. & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ .. \\ .. \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ .. \\ .. \\ u_{n-1} \end{bmatrix}$$

or, more compactly as

$$A.m = U$$

Matrix A is a tridiagonal matrix, that is it has non-zero elements on the three central diagonals only. This set of equations can be solved by Gaussian elimination. In this method the equations are modified so that all values below the leading diagonal computed to be zero. If all of the elements in the first and second rows are divided by the first non-zero elements in each row and then first row is subtracted from the second then the first element in the second row is then zero. This process is repeated for all of the rows. Since the last row will only contain one value this equation can be solved directly which enables the solution of the preceding rows. In this way the whole system of equations can be solved.

4.7 Parametric Representation [11].

Cubic splines can be calculated provided $x_0 < x_1 < \ldots < x_n$, that is there is a unique ordinate for each value of x. Road alignments do not usually satisfy this condition since they may be orientated in any direction. Initially a program was written which rotated the road centre line so that it was parallel to the x-axis but this did not always satisfy the condition along the entire road length. It would have been necessary in some instances to split the alignment into shorter sections that would then require piecing together after calculating the splines.

This condition is too restrictive for road alignments and for general data of this type it is more usual to parameterise the spline by curve length. Since initially the curve length is unknown the cumulative straight line distance *ch* between the data points is often taken to be a sufficient approximation. The curve is then calculated by solving the matrix equation $\mathbf{A.m=u}$ for (*ch,x*) and then for *(ch,y)*.

Radius of Curvature.

The radius at any point on the curve can be calculated from the expression,

$$radius = \frac{(S'_k(x)^2 + S'_k(y)^2)^{1.5}}{(S'_k(x)S''_k(y) + S'_k(y)S''_k(x))}$$

A program CUBSCR.CPP was written to calculate the cubic spline that interpolates a given set of data points and outputs the coordinates of the spline at a specified chainage interval. CUBRAD.CPP calculates the radius of the spline at the same chainage points. These programs were written in Turbo C++ since it is a portable language and the algorithms could be incorporated into other applications.

Initially a program was written to extract the digitised centre line coordinates from a *dxf* file and produce a standard text file that could be read by CUBSCR.CPP. The output from CUBSCR.CPP is in the form of an Autocad script file which is a text file containing commands that enable the curve to be plotted in Autocad. CUBSCR.CPP is listed in Appendix C .

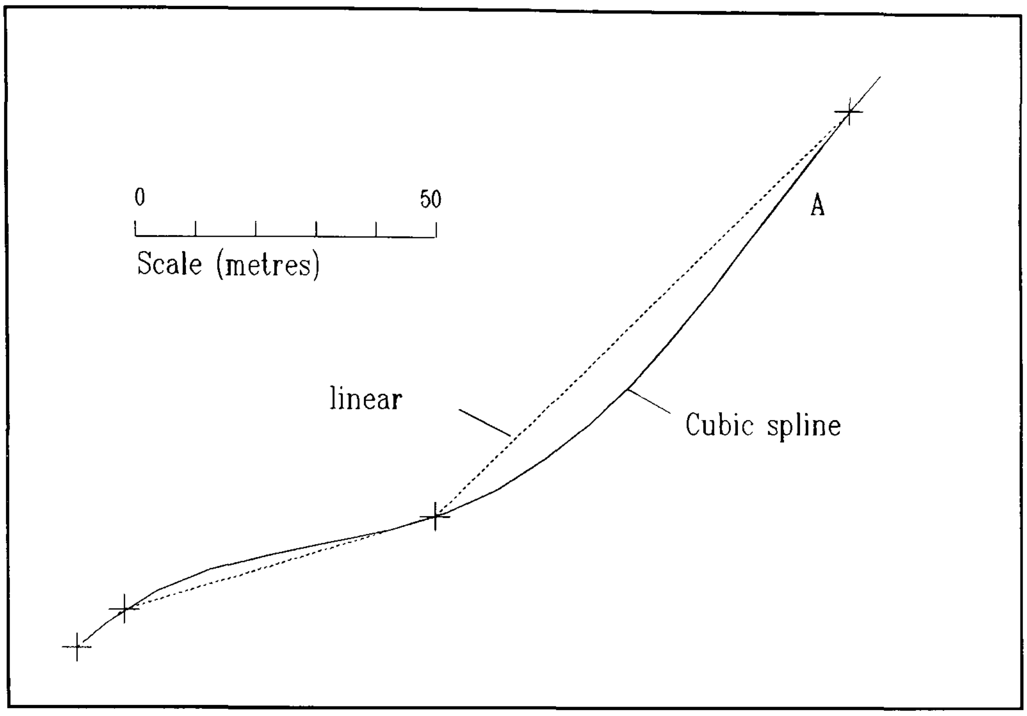


Figure 4.4 Unwanted inflection at A

Cubic splines were fitted to a number of roads using these programs and the resulting curves plotted in Autocad. One disadvantage of the cubic spline is its tendency to insert unwanted points of inflection where none are suggested by the data as shown in Figure 4.4. These points of inflection can be removed by the use of a *shape preserving* spline, that is, the curve conforms to the shape suggested by the data points. Programs were written to produce such a spline using the properties of the B-spline.

4.8 B-SPLINES [11,12].

The minimum number of intervals required to support a cubic spline is four. The cubic spline which exists only for these five abscissas is known as a basis or B-spline. A curve may be constructed which is a linear combination of such cubic splines. At any point the ordinate on the curve will be equal to the sum of the ordinates of the basis functions. For example, in figure 4.5, the ordinate at k_3 is the sum of the ordinates for B_0, B_1 and B_2.

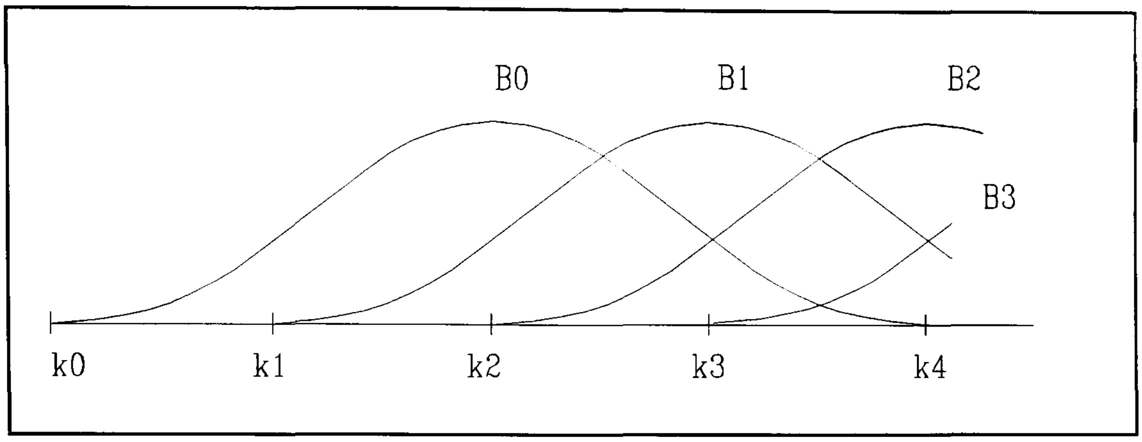


Figure 4.5 B-splines

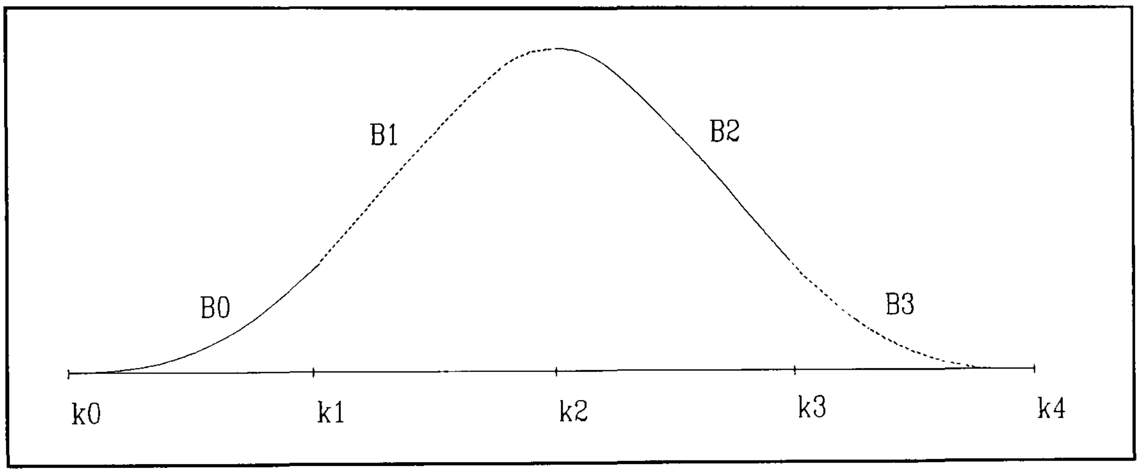


Figure 4.6 Four polynomial segments of the cubic B-spline

The sum of the ordinates at every abscissa is a constant and for a *normalised* B-spline this sum is equated to 1.

$$\sum_{i=-3}^{N+3} B_{i,n}(x) = 1$$

A B-spline made up of four polynomial segments is shown in figure 4.6. The point at which two segments join is known as a *knot*. The equation for each segment is given by,

$$B_i(k) = a_i + b_i k + c_i k^2 + d_i k^3$$

where for a cubic curve $i = 0,1,2,3$. Hence there are four equations and four unknowns. If it assumed for convenience that the knots are placed one unit apart then the equations can be solved from the following conditions:

1. At the ends of the curve the ordinate, slope and curvature are equal to zero.
2. $B_i(1) = B_{i+1}(0)$. The ordinates for two segments are the same at a knot.
3. $B'_i(1) = B'_{i+1}(0)$. The slopes are equal.
4. $B''_i(1) = B''_{i+1}(0)$. The curvatures are equal.
5. $B_0(0) + B_1(0) + B_2(0) + B_3(0) = 1$. Normalised spline.

The solution is

$$B_0(k) = k^3/6$$

$$B_1(k) = (1 + 3k + 3k^2 - 3k^3)/6$$

$$B_2(k) = (4 - 6k^2 + 3k^3)/6$$

$$B_3(k) = (1 - 3k + 3k^2 - k^3)/6$$

The function for the curve is obtained by multiplying each unit B-spline by the value of the data points. This function is calculated from the following equation

$$S(k) = \begin{bmatrix} k^3 & k^2 & k & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i+1} \\ P_i \\ P_{i-1} \\ P_{i-2} \end{bmatrix} /6$$

where $P_i = (x_i, y_i)$.

These equations are incorporated into the programs B_SCR.CPP (Appendix A) and B_RAD.CPP.

The shape of the resulting curve is determined by the polygon formed by the data points, known as the *Bezier Polygon.* It does not, however, interpolate the data points as in the case of the cubic spline. (See figure 4.8). The B-spline does not have the unwanted points of inflection of the cubic spline, that is it is *shape preserving*, and also produces a better estimate of radius.

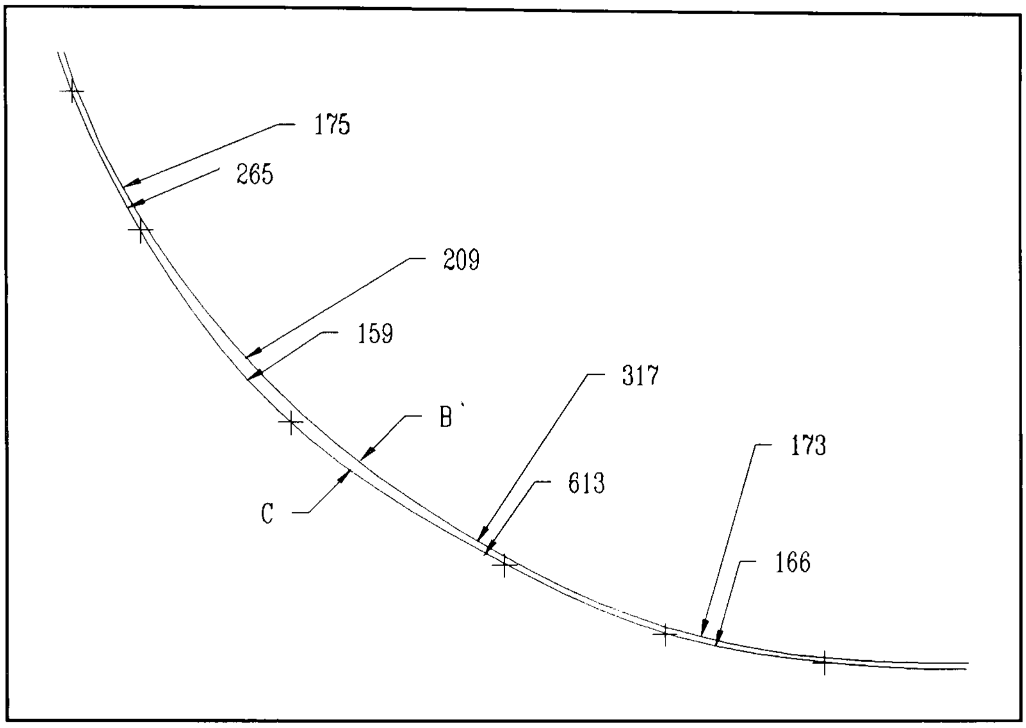


Figure 4.7 Comparison of radii for the cubic and B-spline

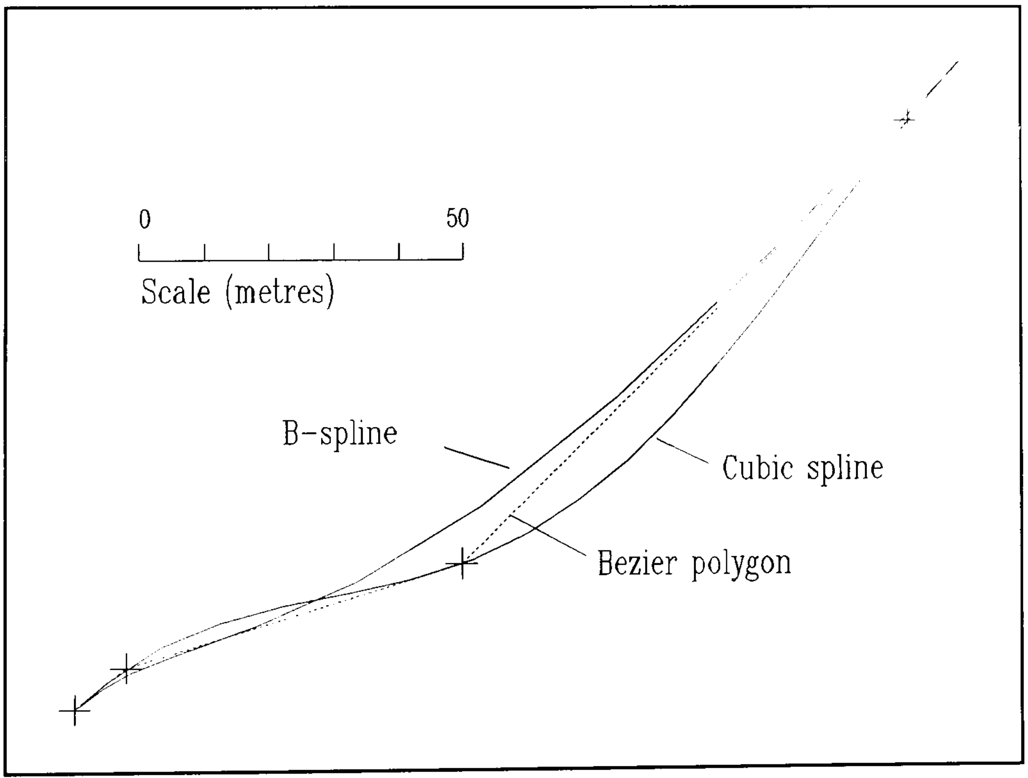


Figure 4.8 Comparison of cubic and B-spline

The short section of the A417 in Oxfordshire follows a 90 degree bend on a constant radius of 210m approx. Figure 4.7 shows a comparison between the cubic and B-spline for part of the alignment. The radius of the cubic spline has a greater variance which is due to inaccuracies in digitising. Even from a large scale plan it is difficult to guarantee an accuracy better than 0.5 metre and variations in the road width can make the position of the centre line difficult to ascertain. The disadvantage of the B-spline is its tendency to flatten the curve at sharp changes in curvature as shown in figure 4.8. This is an important error since it is these bends that are likely to be significant in terms of accidents.

B-splines of order other than cubic were experimented with. Higher orders tend to straighten and smooth the curve and increase the error in the calculated radius. The quadratic B-spline offers a closer fit to the data points than the cubic B-spline, but in practice there was found to be little noticeable difference between the two.

4.9 Smoothing Spline [11,12]

The smoothing spline represents a compromise between the data interpolation of the cubic spline and the shape preserving properties of the B-spline. The main source of error for these splines is inaccuracy in the data. The smoothing spline accounts for these errors by using a weighted least squares fit. Highway engineers are familiar with the use of piano wire for the design of road alignments. The natural shape formed by the wire is a result of minimising its internal energy due to bending. The interpolating function which satisfies this condition is the natural cubic spline. The provision of weighted least squares into the expression enables the removal of unwanted points of inflection by smoothing the data. The weighted least squares expression is

$$E_w(f) = \sum_{j=0}^{N} w_j \left(f(k_j) - f_j\right)^2$$

where f is the approximating function and f_j are the observations with weights $w_j = [1/w_0, \ldots, 1/w_N]^T$. The weights control the degree of data smoothing that takes place.

The minimising function is

$$K_w(f) = J(f) + E_w(f)$$

where $J(f)$ represents the energy due to bending. If the weighting function is zero then a natural cubic spline results and if it is large relative to $J(f)$ the a straight line regression results. The matrix equation for the cubic spline can be rewritten

$$A.M = Df$$

After minimising $K_w(f)$ this function can be modified to

$$(Dw^{-1}D^T + A)M = Df$$

The modified values of the ordinates are given by

$$y = f - w^{-1}D^TM$$

These expressions are incorporated in the program REGSCR.CPP, which calculates the curve, see Appendix A, and REGRAD.CPP, which calculates the radius.

Two examples of the cubic spline, B-spline and smoothing spline after experimenting with different weights, are shown in figures 4.9 and 4.10. These figures illustrate the fact that the smoothing spline represents a compromise between interpolation of the data and the shape of curve suggested by the data. The large difference in the value of the weights used for the two curves suggests that it may be necessary to apply different weights at different locations on the same road to produce the best fit. Whilst this is feasible it is not considered practicable due to the large number of data points involved. The part of the major rural road network examined from Kent consisted of 377 km of road digitised by 5010 points. The two situations shown in figures 4.9 and 4.10 represent fairly extreme situations and generally the difference between the curves is not so visually discernible. By experimenting with different weights a compromise could usually be found, and where the fit was poor additional data points were provided.

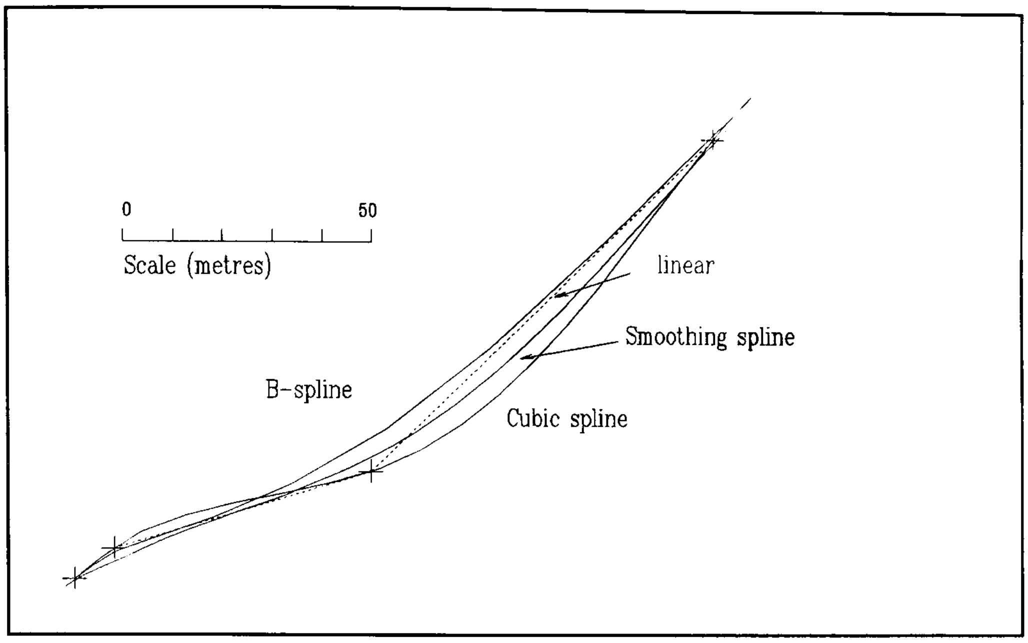


Figure 4.9 Comparison of cubic, B-spline and smoothing spline. weight = 1000

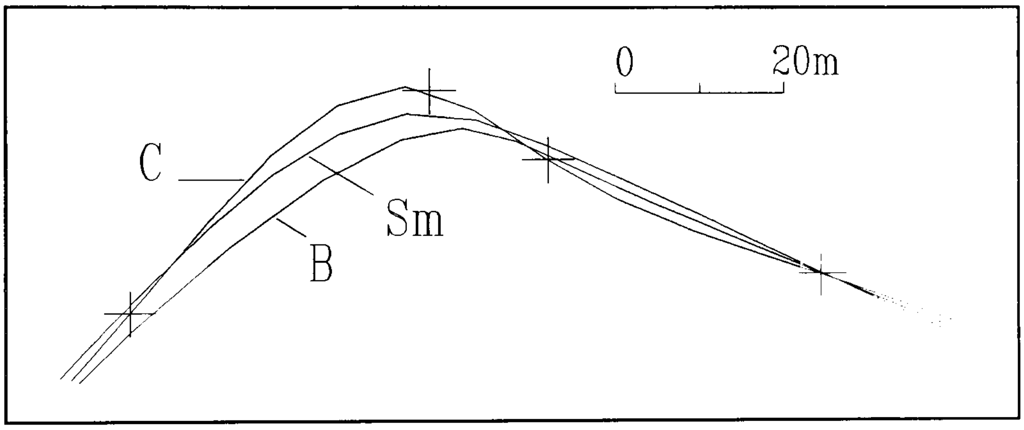


Figure 4.10 Cubic spline, B-spline and smoothing spline

4.10 Shape Preserving Parametrically Defined Curves [13].

The advantage of the B-spline method is that it has shape preserving properties, that is, it retains the shape of curve suggested by the data. The disadvantage is that it is non-interpolating since it is the data points that define the Bezier polygon.

The purpose of this method is to produce a B-spline which interpolates the data. This is achieved by calculating a new Bezier polygon from the data points. For each pair of data points the locations of three additional Bezier points are calculated by the methods outlined below. For the new Bezier polygon a B-spline may be constructed which is tangential to the polygon at each data point. This curve is referred to in this text as the G-spline. A shape preserving curve satisfies the two following conditions.

Local convexity preserving.

If the data points are positively (negatively) convex then the curve is positively (negatively) convex.

Local monotonicity preserving.

If the data points suggest a positive (negative) slope then the curve has a positive (negative) slope. If a local maximum (minimum) is suggested by the data then the curve has a maximum (minimum) in the same data interval.

Suppose $I_i = (x_i, y_i)$, $i = 0,\ldots,N$ are data points in the plane and at each data point there is a tangent vector T_i and $X_i = x_{i+1} - x_i$, $Y_i = y_{i+1} - y_i$,

$$P_i = X_{i-1}Y_i - X_iY_{i-1}$$

If $P_i = 0$ then the points are collinear.

If the polygonal arc suggested by the data is convex then we have

$$T^i = \lambda_i(I_i - I_{i-1}) + (1-\lambda_i)(I_{i+1} - I_i)$$

and

$$\lambda_i = \frac{|P_{i+1}|}{|P_{i+1}| + |P_{i-1}|}$$

The three intermediate Bezier points V^i, J_i and W^i are the calculated for three conditions,

1. $P_iP_{i+1} = 0$. The points are collinear.
2. $P_iP_{i+1} > 0$. The curve is convex.

$$V^i = I_i + \frac{(X_iT_2^{i+1} - Y_iT_1^{i+1})T^i}{(T_1^iT_2^{i+1} - T_2^iT_1^{i+1}) + (X_iT_2^{i+1} - Y_iT_1^{i+1})}$$

$$W^i = I_{i+1} - \frac{(X_iT_2^i - Y_iT_1^i)T^{i+1}}{(T_1^{i+1}T_2^i - T_2^{i+1}T_1^i) + (X_iT_2^i - Y_iT_1^i)}$$

3. $P_iP_{i+1} < 0$. Point of inflection.

$$J_i = \frac{1}{2}(V^i + W^i)$$

$$V^i = I_i + \frac{|X_i T_2^i - Y_i T_1^i| T^i}{|T^i|^2}$$

$$W^i = I_{i+1} - \frac{|X_i T_2^{i+1} - Y_i T_1^i| T^{i+1}}{|T^{i+1}|^2}$$

$$J_i = \frac{1}{2}(V^i + W^i)$$

When this method was used it was found that the inherent errors in the data caused distortions to the curve where sharp changes of direction occurred and also where the data points were in close proximity this caused a congestion of Bezier points. This situation was improved by only calculating the central Bezier point, J_i, see figure 4.11. It was found that this method provides a close approximation to the road alignment but that it is sensitive to data errors.

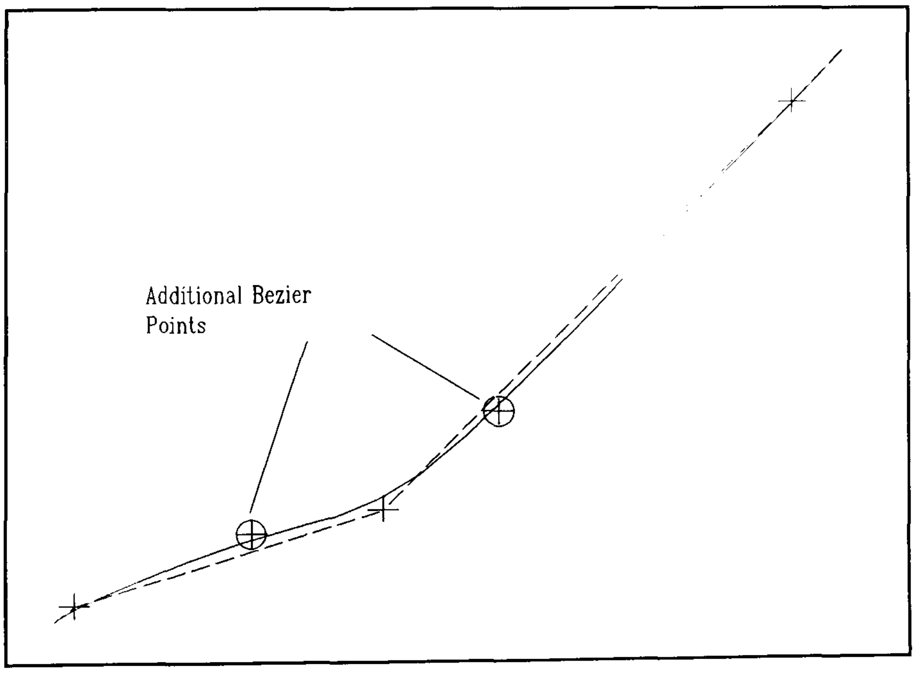


Figure 4.11 G-spline

4.11 Comparison of Spline techniques.

Ultimately the measure of the success of the curve fitting method depends upon the reliability of the calculated radii. Using the data for the circular arc on the A417, shown in figure 4.8, table 4.3 shows the calculated radii and their means and standard deviations for the cubic, B-spline, smoothing spline with weights of 500, 1000, 2000 and 5000, and the G-spline.

The B-spline gives the closest agreement with the estimated radius of 210 metres and the smoothing spline estimate improves with increasing weight. However weights of 2000 and greater are likely to cause inaccuracies elsewhere on the alignment.

	Cubic	Bspline	Smoothing spline				Gspline
			w500	w1000	w2000	w5000	
	198	201	187	188	188	200	356
	215	162	216	215	213	202	115
	265	175	254	248	239	215	322
	235	235	231	227	222	213	288
	205	233	207	206	205	203	163
	159	209	166	169	174	183	159
	148	175	150	155	161	173	215
	215	235	206	206	205	206	163
	278	289	255	248	239	229	202
	613	317	470	410	351	291	876
	282	223	288	283	276	307	280
	289	178	205	212	220	227	109
	128	166	136	146	159	178	292
	166	173	170	174	180	188	130
	231	201	222	214	205	198	141
mean	241	211	224	220	215	214	254
Stdev	110	44	77	62	47	37	183

Table 4.3 Estimated radii, means and standard deviations for the A417 for different spline types.

A section of the A257 in Kent having abrupt changes in radius was also chosen for comparison. (Figure 4.12).

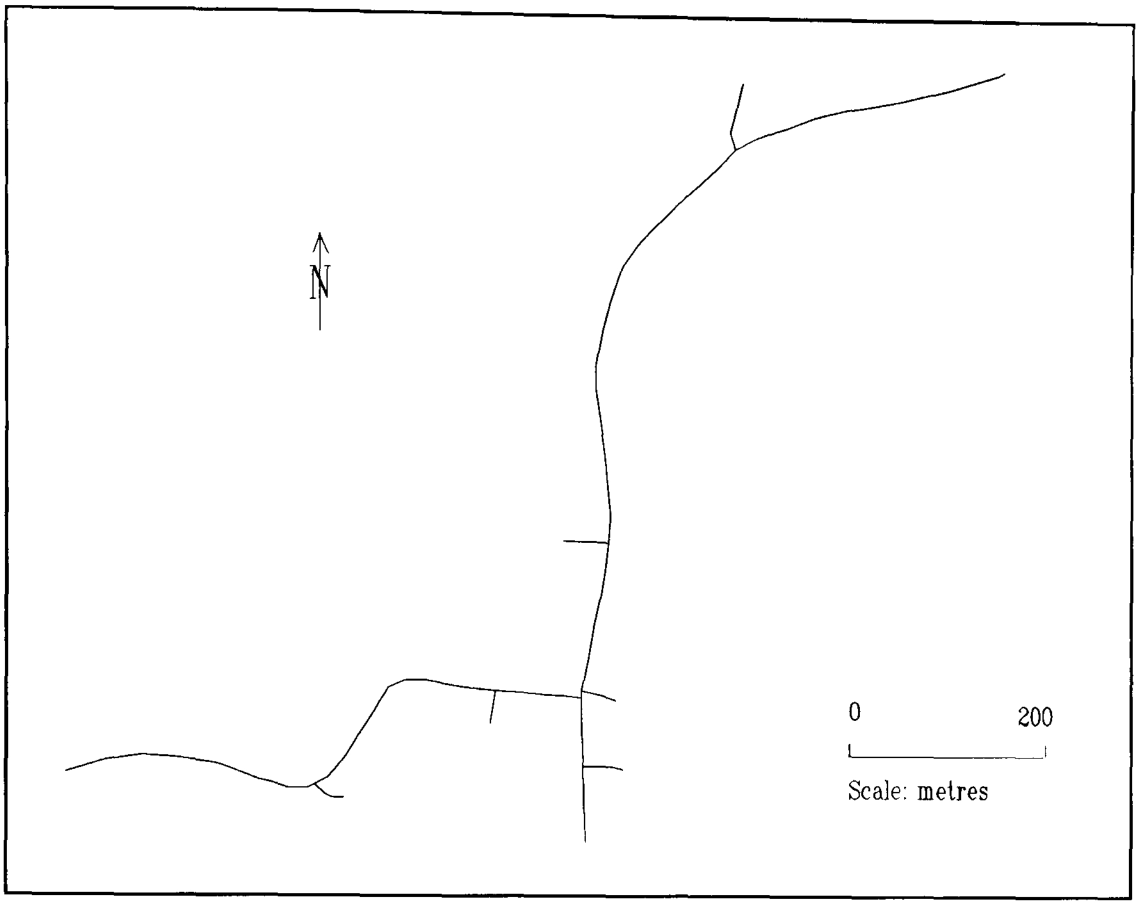


Figure 4.12 A257 at Wingham in Kent

Table 4.4 shows a comparison of radii for the cubic spline, B-spline, G-spline and various weights of smoothing spline. Left hand bends are shown as positive and right hand as negative. If a radius is found to be greater than 5000 metres then the radius is taken to be 5000 metres since for a road alignment this is effectively straight. The G-spline follows the digitised alignment more faithfully than the cubic spline but they both have a tendency to insert unwanted points of inflection. This surprising tendency of the G-spline is due to the inherent errors in the data. This is illustrated at chainages 0.64 and 0.68 where sharp radii of opposite curvature to that at 0.66 can be seen for both curves.

Table 4.5 shows the absolute difference between the curvature of each spline and that estimated from the drawing. The sum of these values, which is also shown, is an indication of their efficiency as estimators of radius. From this table it can be seen that the smoothing spline produces the most favourable result.

CH	DWG	Cubic	Bspline	Smoothing Spline			Gspline
				W500	W1000	W2000	
0.140	-160	-154	-200	-148	-156	-167	-91
0.160	-132	-89	-128	-116	-128	-145	-95
0.180	-160	-511	-268	-500	-404	-326	-120
0.200	5000	821	-431	-1484	-678	-409	-1149
0.220	-190	-174	-230	-161	-170	-180	-1102
0.240	-190	-133	-290	-168	-189	-217	-251
0.260	5000	-909	-921	-2764	-2530	-3756	-159
0.280	5000	197	400	200	242	263	2048
0.300	120	170	150	132	124	112	753
0.320	38	12	100	53	44	46	77
0.340	38	37	53	45	51	52	166
0.360	73	61	100	78	79	80	29
0.380	101	303	265	511	341	235	27
0.400	5000	902	700	1913	877	613	87
0.420	5000	-989	5000	1068	5000	-887	212
0.440	5000	366	366	-233	-154	-122	-314
0.460	-30	-25	-21	-25	-32	-34	-792
0.480	-30	-50	-49	-47	-47	-50	-84
0.500	-100	-89	-120	-152	-132	-122	-29
0.520	5000	-4264	5000	421	510	1019	-51
0.540	5000	268	681	346	431	623	-66
0.560	5000	1477	1180	763	897	1190	931
0.580	5000	-1563	1900	5000	5000	2318	5000
0.600	5000	398	2600	-1178	-1568	4007	1149
0.620	5000	142	2330	-703	-1177	1693	-705
0.640	17	-37	-572	74	60	50	-94
0.660	17	2	3	16	9	18	7
0.680	5000	-32	-1589	139	111	95	-195
0.700	5000	-2234	1100	-339	-478	-1395	479
0.720	5000	720	-4107	-479	-501	-552	1865
0.740	5000	-409	-1524	-1390	-855	-659	-1326
0.760	225	-1252	1799	1860	2642	5000	-900
0.780	225	1136	700	563	516	516	761
0.800	225	412	495	326	320	323	361
0.820	350	263	430	231	240	253	281
0.840	350	190	290	212	224	240	216
0.860	350	320	255	569	542	505	774
0.880	350	-1285	1000	5000	4514	1786	814
0.900	350	-2301	5000	3107	2588	2221	-3273
0.920	5000	2785	5000	1466	1819	2931	-1315
0.940	5000	875	-2400	3881	5000	-3602	5000
0.960	5000	-1027	-1140	-430	-425	-407	-3062
0.980	-200	-254	-492	-202	-208	-213	-530
1.000	-160	-143	-208	-142	-148	-158	-223
1.020	-160	-192	-175	-230	-238	-244	-90
1.040	5000	-522	-322	-545	-506	-442	376
1.060	5000	2788	-377	4636	-2014	-735	-219
1.080	-200	-759	-354	-192	-192	-194	-1071
1.100	-90	-109	-160	-88	-96	-108	-173
1.120	-90	-100	-125	-151	-154	-161	-56
1.140	-90	-262	-208	-364	-322	-287	-612

Table 4.4. A257 at Wingham

CH	Cubic	B-spline	Smoothing Spline			G-spline
			W500	W1000	W2000	
0.140	0.0002	0.0013	0.0008	0.0005	0.0002	0.0043
0.160	0.0050	0.0016	0.0032	0.0024	0.0016	0.0021
0.180	0.0043	0.0025	0.0047	0.0043	0.0038	0.0071
0.200	0.0075	0.0039	0.0064	0.0056	0.0048	0.0072
0.220	0.0005	0.0019	0.0003	0.0000	0.0004	0.0023
0.240	0.0013	0.0028	0.0002	0.0003	0.0010	0.0000
0.260	0.0051	0.0052	0.0064	0.0059	0.0059	0.0067
0.280	0.0113	0.0088	0.0120	0.0112	0.0104	0.0076
0.300	0.0121	0.0129	0.0129	0.0138	0.0143	0.0192
0.320	0.0896	0.0163	0.0251	0.0251	0.0290	0.0123
0.340	0.0333	0.0251	0.0306	0.0285	0.0259	0.0407
0.360	0.0226	0.0163	0.0184	0.0191	0.0189	0.0433
0.380	0.0096	0.0100	0.0075	0.0082	0.0092	0.0177
0.400	0.0074	0.0077	0.0064	0.0068	0.0074	0.0110
0.420	0.0052	0.0064	0.0075	0.0072	0.0064	0.0031
0.440	0.0090	0.0090	0.0056	0.0020	0.0002	0.0050
0.460	0.0338	0.0414	0.0392	0.0338	0.0250	0.0057
0.480	0.0138	0.0142	0.0138	0.0150	0.0150	0.0282
0.500	0.0050	0.0021	0.0001	0.0003	0.0013	0.0134
0.520	0.0060	0.0064	0.0090	0.0086	0.0082	0.0089
0.540	0.0100	0.0077	0.0093	0.0091	0.0086	0.0073
0.560	0.0069	0.0071	0.0073	0.0076	0.0074	0.0064
0.580	0.0056	0.0068	0.0064	0.0064	0.0064	0.0071
0.600	0.0088	0.0066	0.0059	0.0054	0.0056	0.0048
0.620	0.0133	0.0067	0.0053	0.0048	0.0054	0.0044
0.640	0.0208	0.0045	0.0133	0.0198	0.0229	0.1491
0.660	0.5062	0.3396	0.0589	0.0688	0.1174	0.0011
0.680	0.0250	0.0056	0.0113	0.0134	0.0153	0.0083
0.700	0.0058	0.0072	0.0038	0.0033	0.0042	0.0068
0.720	0.0076	0.0060	0.0045	0.0042	0.0043	0.0070
0.740	0.0038	0.0056	0.0057	0.0055	0.0051	0.0051
0.760	0.0055	0.0068	0.0068	0.0068	0.0066	0.0076
0.780	0.0071	0.0077	0.0079	0.0080	0.0082	0.0090
0.800	0.0087	0.0083	0.0093	0.0093	0.0094	0.0098
0.820	0.0101	0.0086	0.0107	0.0106	0.0104	0.0109
0.840	0.0115	0.0097	0.0112	0.0110	0.0107	0.0075
0.860	0.0094	0.0102	0.0079	0.0080	0.0081	0.0075
0.880	0.0055	0.0073	0.0060	0.0064	0.0065	0.0066
0.900	0.0058	0.0064	0.0065	0.0066	0.0066	0.0070
0.920	0.0066	0.0064	0.0070	0.0069	0.0068	0.0064
0.940	0.0074	0.0058	0.0067	0.0065	0.0064	0.0066
0.960	0.0053	0.0054	0.0040	0.0039	0.0039	0.0044
0.980	0.0023	0.0042	0.0012	0.0013	0.0014	0.0018
1.000	0.0007	0.0014	0.0009	0.0008	0.0005	0.0049
1.020	0.0010	0.0005	0.0017	0.0019	0.0020	0.0089
1.040	0.0043	0.0031	0.0045	0.0044	0.0043	0.0017
1.060	0.0066	0.0036	0.0071	0.0065	0.0058	0.0072
1.080	0.0049	0.0034	0.0010	0.0010	0.0010	0.0005
1.100	0.0029	0.0000	0.0058	0.0051	0.0042	0.0116
1.120	0.0037	0.0017	0.0005	0.0004	0.0002	0.0046
1.140	0.0024	0.0014	0.0038	0.0035	0.0031	0.0045
Σ	1.0082	0.7007	0.4524	0.4556	0.4971	0.5820

Table 4.5 Absolute differences in curvature.

An additional interpolating method was considered and is outlined below but was found to be inappropriate in this situation.

4.12 Rational Interpolants in Tension [14].

The rational interpolant considered is a spline with a cubic numerator and a quadratic denominator which includes a tension parameter C_k. The inclusion of the tension parameter is analogous to pulling on the two ends of the curve. A value of C_k close to zero will produce a linear spline and high values will tend to produce a cubic spline. The advantage of this method is that it can remove unwanted points of inflection. However, equations of this form cannot be parameterised by curve length and so can only used where x is single valued. This makes them unsuitable for general data, such as road alignments, but they do provide an opportunity to examine the effect of tension on an interpolating spline. The function is defined by the formula

$$S_k(x) = y_k(1-w)+y_{k+1}w - C_k h_k(1-w)w\frac{A_k(1-w)+B_k w}{C_k+h_k^2(1-w)w}$$

where $h_k=x_{k+1}-x_k$, $w=(x-x_k)/h_k$, $d_k=(y_{k+1}-y_k)/h_k$,

$$A_k = d_k-m_k,\ B_k = m_{k+1}-d_{k+1} \text{ and } m_k = S'(x_k).$$

Sections of road were selected that satisfied the condition $x_k <= x <= x_{k+1}$ and $k = 0,1,\ldots, n-1$. It was found that the required degree of tension varied along the alignment. Large radius curves tended to require high values of C_k but low radius curves required low values of C_k. This fact combined with its non parametric form makes it impracticable as a solution in this situation.

4.13 Conclusions

The use and availability of digitised road alignments have made it feasible to obtain more precise information about the road geometry than was previously available. By using polynomial splines it is possible to gain a reasonable estimate of the centre line radius. The accuracy of those estimates, and to an extent the method used, is highly dependent upon the accuracy and placement of the digitised points. Common sources of error were found to be, too few points at sudden changes in curvature and uneven placement of the data points. On straight sections of road there is a tendency to provide fewer data points that allows the spline too much freedom to deviate from the centre line.

The best estimates of radius for the available data were provided by the smoothing spline after experimenting with different weights. In some instances it was found necessary to provide additional data points in order to constrain the spline. Fitting methods such as the G-spline which accurately follow the shape of curve suggested by the data do not necessarily provide the best estimate of radius due to their tendency to amplify data errors that are inherent in the digitising process. It should also be noted that the methods used were those found appropriate for the digitised alignments that were available. For another set of alignments a different curve fitting procedure could be appropriate.

This discussion of spline fitting procedures has not exhausted all of the methods currently available but it is felt that any method will ultimately be limited by the accuracy of the data. This could be improved by using centre line data from the Ordnance Survey rather than data digitised from maps. The priority for most spline fitting techniques is to provide a smooth curve which interpolates the data with little emphasis on the accuracy of the radius. In the case of a road alignment the radius is of fundamental importance and there may be scope for developing a suitable spline method with this point in mind.

Chapter 5
Accidents and Horizontal Curvature

5.1 Introduction

After fitting cubic splines to a digitised alignment an investigation was carried out to investigate the relationship between the road centre line radius and the expected number of accidents. Most of the roads for which accident data was supplied from Kent were rural single carriageway A roads and it was these that were used for analysis. Relationships of this type have been investigated before [15,16]; however, in most cases the centre line data has been either extracted manually from a plan or, by site survey. In this study, it is extracted automatically from the database which has made a large volume of data available for analysis.

The regression spline technique, using a weight of 1000, was used since this produced the best overall fit. In the case of a spline the quality of fit is largely subjective and depends upon the requirements for the resulting curve. In this case the requirement was that the spline should provide a good estimate of radius, especially for low radius curves that are likely to be the most dangerous.

5.2 Data

The splines were calculated at 10 metre intervals and the radius was calculated at each point. The accidents were also located to the nearest 10 metres and so an estimate of radius was available at each accident location. From the database, accidents and centre line points were selected for rural A roads for the six year period from 1984 to 1989 inclusive. Junction accidents were eliminated since the proximity and geometry of a junction is likely to have a more significant effect on accidents than the centre line radius.

Since the radius of a spline is continuously changing, unlike a circular arc, it was not possible to locate bends of constant radius greater than 10 metres in length. The relationship between accidents and curvature was investigated by comparing the frequency distributions of the accident radii, categorized at 50 metre intervals, with the frequency of the chainage points at the same intervals. Initially it was intended to consider a linear model and, since accidents are likely to be inversely related to radius, the relationship between the number of accidents and curvature was investigated. The analysis was carried using generalised linear models.

5.3 Generalised linear models [17,18]

Generalised linear models can be used to solve a variety of statistical problems involving linear combinations of parameters. A set of observations Y_i can be modelled as a set of random variables, with Y_i having a probability distribution with mean μ_i . For a generalised linear model the probability distribution for Y_i is taken from the exponential family of distributions.

The linear combination of the explanatory variables is known as the systematic part of the model and is given by

$$\eta_i = \sum_j x_{ij} \beta_j \ ,$$

η is a function of the expected value of Y,

$$\eta_i = g_i(\mu_i) \ ,$$

and the function g is known as a *link function* since it links the systematic part of the model to the probability distribution of Y.

The parameters of the statistical model are estimated by the method of *maximum likelihood*, that is, by calculating the values of the parameters that maximise the probability of the observed values of Y. The calculations were carried out using the computer program GLIM.

5.4 Poisson models

The recorded number of accidents at a site Y is usually assumed to be a random variable with a Poisson distribution. The distribution depends upon the parameter λ the mean value of y,

$$f(y;\lambda)=\frac{\lambda^{y}}{y!}e^{-\lambda}$$

Two forms of the link function were used for the analysis, the identity function

$$\eta = \lambda\ ,$$

and the log link function

$$\eta = \log\lambda\ ,$$

5.5 Accidents and curvature

The relationship between the expected number of accident and curvature was investigated for radii less than 2000 metres categorized at 50 metre intervals providing 40 observations. For practical purposes radii of 2000 metres or above can be considered straight from a driver's point of view. An allowance was made for the fact that the length of road for each category of radius varied. The response variable Y was taken to be the number of accidents per kilometre at a given radius. However, when the number of accidents is divided by length the error structure is no longer Poisson since the variance is no longer equal to the mean. GLIM allows for this if the length is applied as a weight. Two models were used, the linear model

$$\mathrm{E}(Y) = \alpha + \beta\kappa$$

and the exponential model,

$$\log \mathrm{E}(Y) = \alpha + \beta\kappa$$

where $\mathrm{E}(Y)$ is the expected number of accidents per kilometre,
κ is the curvature, and α and β are the parameters of the model.

A comparison of the two fitted models is shown in table 5.1

model	α	β	deg of freedom	Scaled deviance	chi-squared
linear	4.854	613.9	38	42.357	43.61
exponential	1.699	51.23	38	45.850	46.59

Table 5.1 Comparison of linear and exponential models

The scaled deviance is an indication of the goodness of fit. It is given by,

$$D = 2[l(y;y) - l(\hat{\mu};y)]$$

where $l(\hat{\mu};y)$ equals the log-likelihood function. In this case $\hat{\mu}_i$ is the fitted value obtained by substituting the maximum likelihood estimates $\hat{\alpha}$ and $\hat{\beta}$ in the appropriate model and so $l(\hat{\mu};y)$ is the maximised log-likelihood. $l(y;y)$ is the maximum log likelihood where there is one parameter for each observation. An alternative measure of goodness of fit is the ordinary, or Pearson, chi-squared statistic.

$$X^2 = \sum \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}$$

If the model is satisfactory then both the scaled deviance and the Pearson chi-squared have chi-squared distributions with 38 degrees of freedom. The degrees of freedom being the number of observations minus the number of parameters estimated. In table 5.1 none of the values are significant indicating that both models provide a satisfactory fit and that the linear model fits slightly better than the exponential model. However, this is not the only criterion for the adequacy of the models. Figure 5.1 shows the two models and the observed values at low radii. At low radii the expected number of accidents for the exponential model approaches infinity more rapidly than the linear model. This may lead to unrealistically high estimates of the number of accidents at low radii for the exponenetial model.

Figure 5.2 shows the linear model. It can be seen that the influence of radius on the expected number of accidents diminishes for radii over 600 metres.

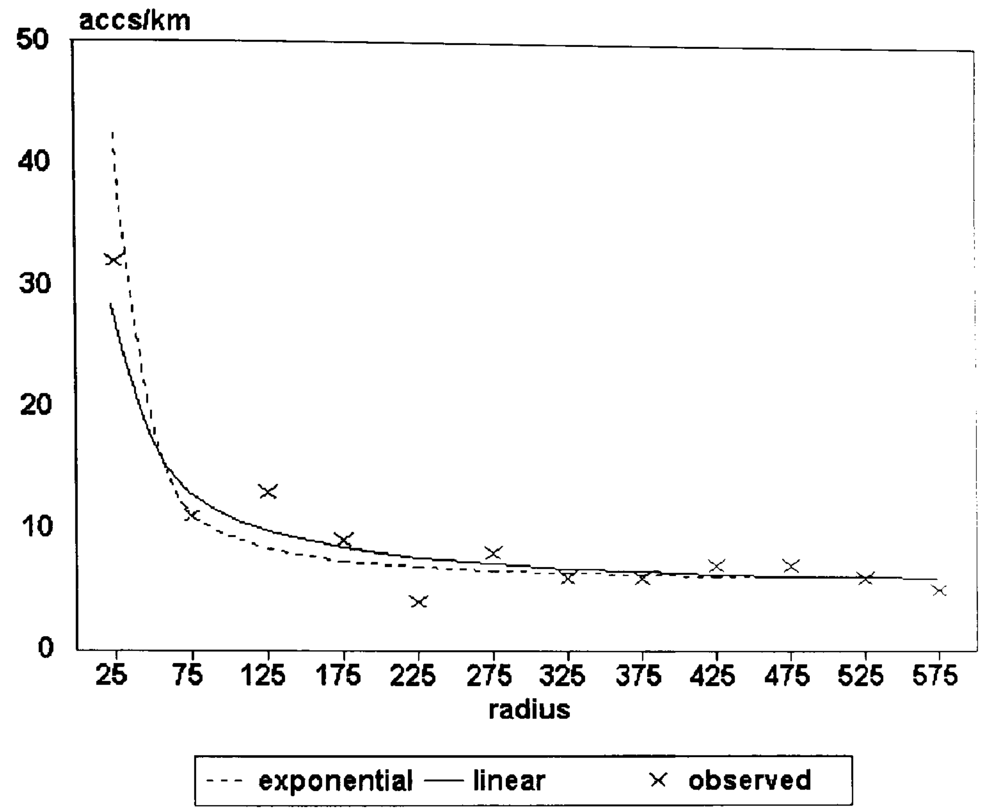


Figure 5.1 Comparison of models at low radii

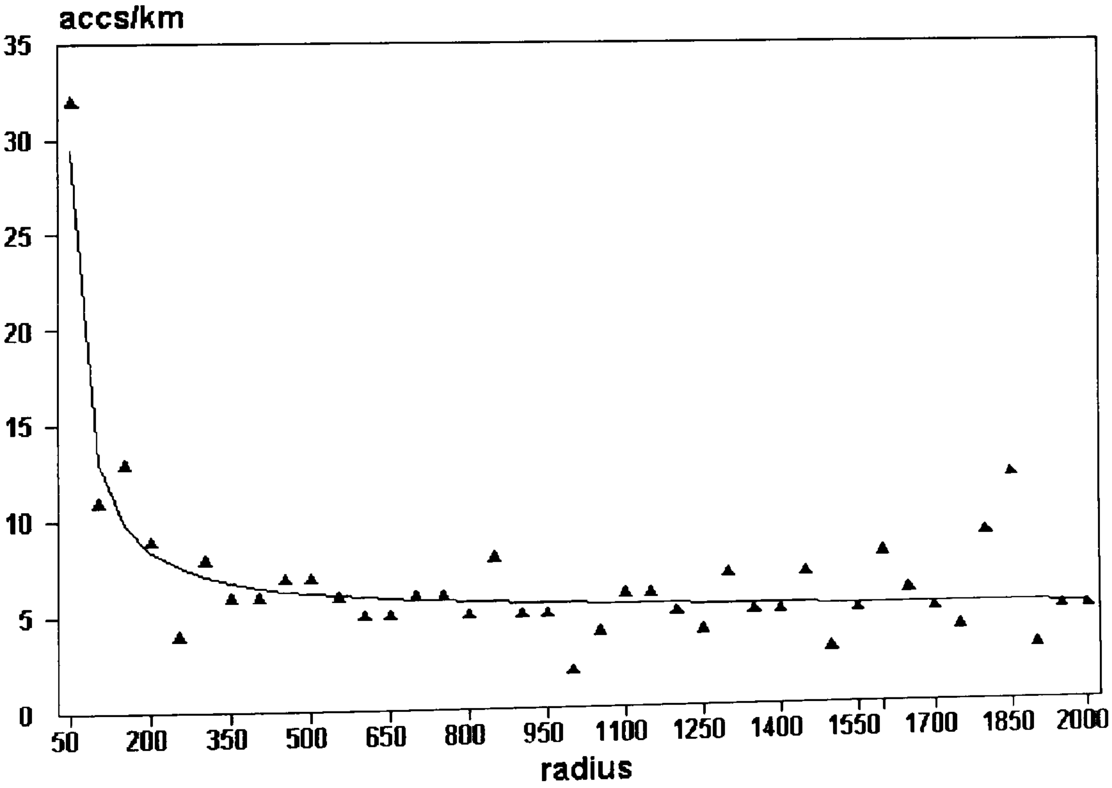


Figure 5.2 Linear Poisson model

5.6 Traffic flows

The above models do not contain traffic flow as an explanatory variable. As the volume of traffic increases on a section of road then more vehicles are exposed to the risk of an accident. The mean number of vehicles per annum over the six year period was calculated for each link. For each individual year the number of vehicles is the average annual daily flow×365.

Four models were compared using GLIM: the two previous models without traffic flow fitted to the new data set and two models incorporating traffic flow, linear and exponential. The linear model has the form

$$E(\frac{Y}{q}) = (\alpha + \beta\kappa).$$

This can be fitted in the same way as the previous linear model, except that Y is replaced by Y/q and the weight is length multiplied by flow. The exponential model is

$$\log E(Y) = \alpha + \beta\kappa + \gamma \log q$$

where q = traffic flow in vehicles per year.

Of the four models the exponential model incorporating traffic flow has the lowest scaled deviance and chi-squared value indicating the best fit; see table 5.2. However, the values of chi-squared are all fairly high and particularly high for the linear model, throwing some doubt on the models. Figure 5.3 shows the linear model with traffic flows in the range 2 to 3 million vehicles per annum only, for reasons of clarity. The graph shows a wide scatter of observed values which is partly due to the categorisation of the data by both radius and traffic flow. This has caused relatively short lengths of road in each category which become apparent when the observed number of accidents is divided by length. As an alternative, models were fitted with the data categorized by curvature as opposed to radius but this tended to exacerbate the problem, particularly on low radius bends.

model	α	β	γ	deg of freedom	Scaled deviance	chi-squared
linear	4.857	612.9	0	967	1027	1396
exponential	1.700	51.22	0	967	1031	1337
linear	1.212	258.7	1	967	973	1507
exponential	0.989	58.28	0.589	966	945	1236

Table 5.2 Comparison of models without and with traffic flow

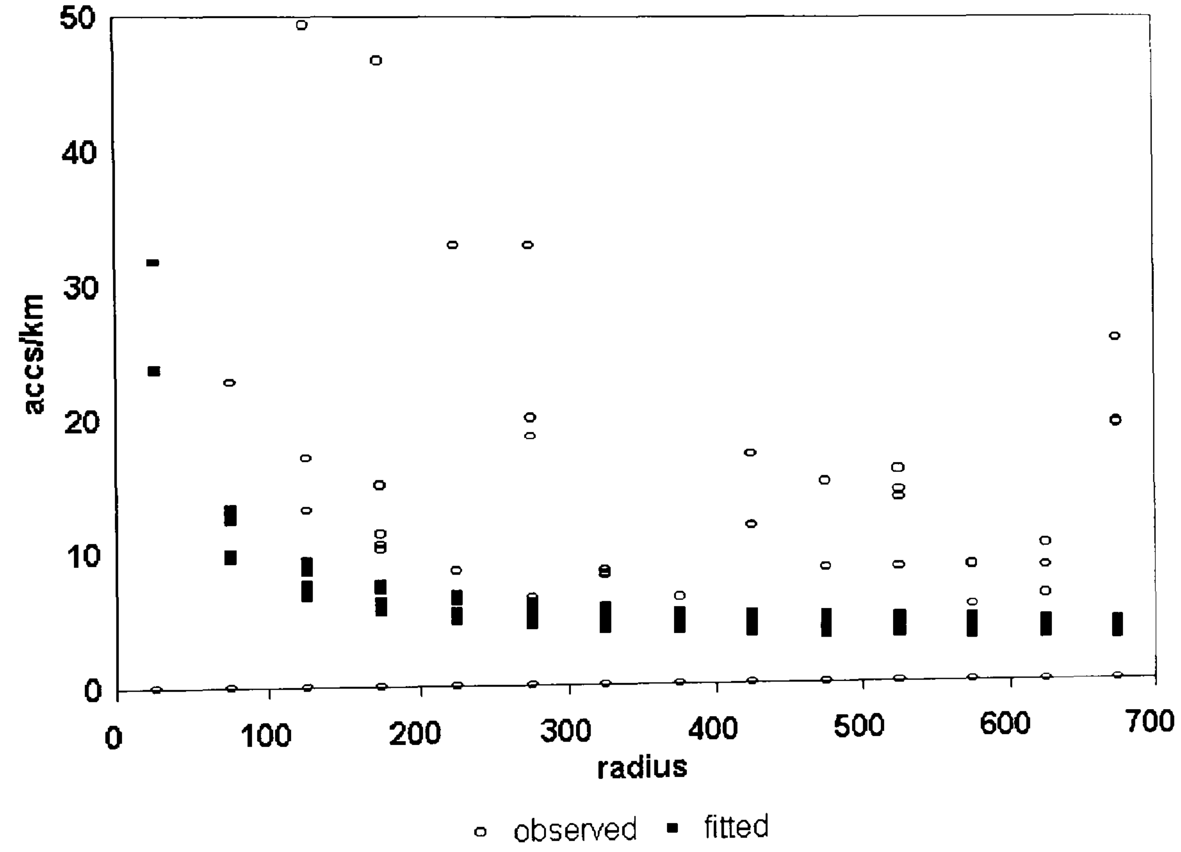


Figure 5.3 Expected accidents per km in the range 2 to 3 million vehicles per year

There are also other variables which have not been included in the models, such as the vertical alignment, sight distance and the length of the bend. Whilst accidents may tend to occur at the apex of the bend there is no indication of the abruptness of the bend. A longer transition on the approach would tend to improve safety. There is also no indication to the general context of the road alignment: a bend having a particular radius may have fewer accidents on a winding road than on a road which is otherwise straight.

There are three main sources of error in the data: the location of the accident, the accuracy of the estimate of radius and the traffic flow. The categorisation of the data by radius tends to reduce the error in the geometry, but to account for the error in traffic flow a more comprehensive coverage of traffic counts would be required.

5.7 Conclusion

The use of digitised road alignments can potentially make a large amount of data relating to road geometry available automatically. By fitting cubic splines to the data it is possible to provide an estimate of curve radius. This chapter examined the possibility of using this data to investigate the relationship between the expected number of accidents, curvature and traffic flows. Whilst there is firm evidence that both curvature and traffic flow influence the number of accidents the models were only partly satisfactory. The method shows, however, sufficient potential for application to a larger and more comprehensive data set.

After investigating the method of representing the centre line in the database, the next consideration is the network configuration itself.

Chapter 6
Network definition

In chapter 3 it was shown that the inclusion of the network definition in the accident tables was a source of redundancy. In this chapter a system is devised for defining the network in the database and its advantages are demonstrated by extracting the number of accidents occurring at minor junctions.

6.1 Link and Node System

The link and node system has been adopted for the network definition since it is an appropriate system for many types of network analysis. For accidents it also has the advantage that by attributing them to individual links they form relatively small blocks of data. In the systems where they are attributed to the entire road, the data blocks become large which in turn makes search operations less efficient. Whilst standard programs are not available for the spatial analysis of accidents, for example, for accident migration, the link and node system would provide an appropriate framework. To retain the universal concept of GIS then as far as possible a single definition of the road network for all traffic management purposes is preferable.

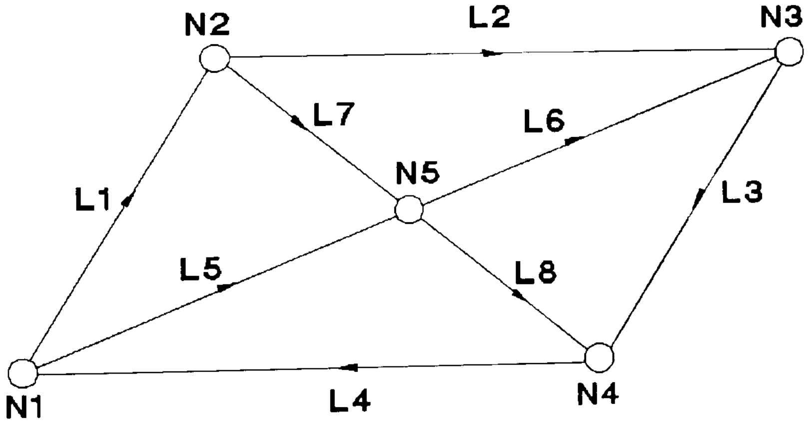


Figure 6.1 Network example

Figure 6.1 shows a network consisting of five nodes and eight links. In this network a junction is considered to be a point. The arrows show the direction of chainage for each link.

The location of each link is defined by a series of eastings and northings which can be stored in a GIS, see table 6.1. The chainage of these points can be easily calculated, see chapter 4. Table 6.2 shows the link definition table, which includes the link number, the road number, the link length which was calculated in table 6.1, the nodes at each end of the link and also the time period for which this link is in existence within the database.

LINK	EASTING	NORTHING	CHAINAGE
L1	x_0	y_0	0.00
L1	x_1	y_1	ch_1
L1	....	....	....
L1	x_k	y_k	ch_k
L2	x_k	y_k	0.00
L2	x_{k+1}	y_{k+1}	ch_{k+1}

Table 6.1 Link location

LINK	ROAD	LENGTH	SN	EN	SD	ED
L1	A1	*len1*	N1	N2	01/01/80	06/06/94
L2	A1	*len2*	N2	N3	01/01/80	06/06/94
L3	A2	*len3*	N3	N4	01/01/80	06/06/94
L4	A2	*len4*	N4	N1	01/01/80	06/06/94
L5	A3	*len5*	N1	N5	01/01/80	06/06/94
L6	A3	*len6*	N5	N3	01/01/80	06/06/94
L7	A4	*len7*	N2	N5	01/01/80	06/06/94
L8	A4	*len8*	N5	N4	01/01/80	06/06/94

Table 6.2 Link Table

6.2 Connecting the links and nodes

The connectivity of the network can be established by a suitable query of this table. The following SQL program carries this out:

```
select sn as node, link, road, 'start' as flag from link ;
union select en, link, road, 'end' from link ;
order by 1 ;
into table connect
```

The output is shown in table 6.3.

NODE	LINK	ROAD	FLAG
N1	L1	A1	start
N1	L4	A2	end
N1	L5	A3	start
N2	L1	A1	end
N2	L2	A1	start
N2	L7	A4	start
N3	L2	A1	end
N3	L3	A2	start
N3	L6	A3	end
N4	L3	A2	end
N4	L4	A2	start
N4	L8	A4	end
N5	L5	A3	end
N5	L6	A3	start
N5	L7	A4	end
N5	L8	A4	start

Table 6.3 Nodes and connected links.

The flag 'start' or 'end' shows whether the node is at the start or end of a link so distances relative to the node can be calculated in terms of the chainage of the link.

Table 6.4 shows the node table. The chainage of the node on the links that it connects can be found via an SQL program from table 6.2 and its coordinates from tables 6.1 and 6.2. The road number in the node table is the major road which is the road that accidents are usually assigned to. The number of arms is counted from table 6.3. The number of arms and the junction type would enable the COBA classification to be determined.

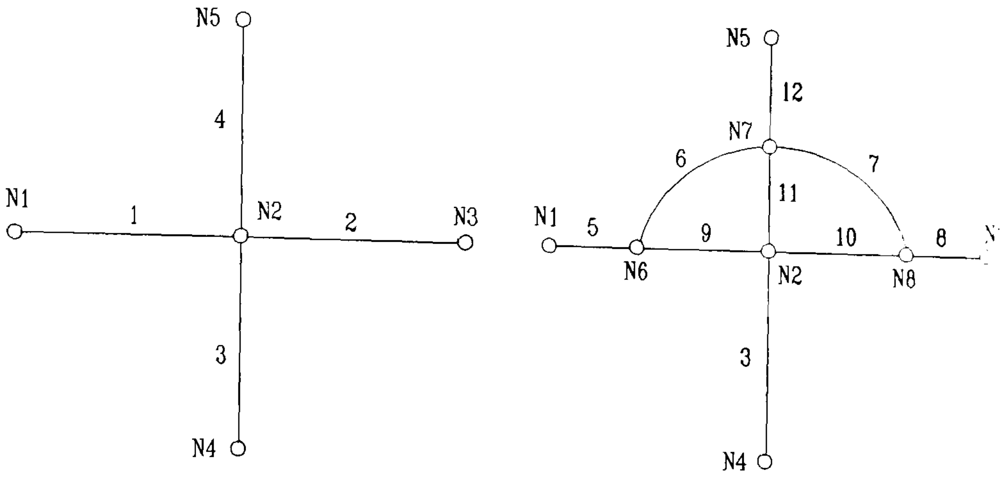
NODE	E	N	ROAD	ARMS	TYPE	SD	ED
N1	*x*	*y1*	A1	*3*	TSG	01/01/80	06/06/94
N2	*x2*	*y2*	A1	*3*	TSG	01/01/80	06/06/94
N3	*x3*	*y3*	A1	*3*	TSG	01/01/80	06/06/94
N4	*x4*	*y4*	A2	*3*	TSG	01/01/80	06/06/94
N5	*x5*	*y5*	A3	*4*	RBT	01/01/80	06/06/94

Table 6.4 Node table

6.3 Bypasses

The road network is in a constant state of change with nodes and links being added and realigned continually. It is important therefore that the database should take changes to the network into account.

Figure 6.2 shows the modification to the network caused by the opening of a bypass on 01/01/89. The nodes and link 3 in figure 6.2(i) are unaffected by the introduction of the bypass and are unchanged in figure 6.2(ii). Three new nodes are added, *N*6, *N*7, and *N*8 and two new links 6 and 7. Links 1,2 and 4 are terminated when the bypass is opened and are replaced by links 5 and 9, 10 and 8, and 11 and 12 respectively. The resulting link table is shown in table 6.5. The road *A*1 now follows the line of the bypass and the displaced *A*1 is reclassified as the *C*1. It is also apparent from this network that the start and end dates for the nodes cannot be determined from the link table since they are not necessarily the same, as in the case of *N*1 which is now associated with two links of different time periods.



(i) (ii)

Figure 6.2 Addition of bypass

LINK	ROAD	LENGTH	SN	EN	SD	ED
L1	A1	*len1*	N1	N2	01/01/80	31/08/89
L2	A1	*len2*	N2	N3	01/01/80	31/08/89
L3	A2	*len3*	N4	N2	01/01/80	06/06/94
L4	A2	*len4*	N2	N5	01/01/80	31/08/89
L5	A1	*len5*	N1	N6	01/09/89	06/06/94
L6	A1	*len6*	N6	N7	01/09/89	06/06/94
L7	A1	*len7*	N7	N8	01/09/89	06/06/94
L8	A1	*len8*	N8	N3	01/09/89	06/06/94
L9	C1	*len9*	N6	N2	01/09/89	06/06/94
L10	C1	*len10*	N2	N8	01/09/89	06/06/94
L11	A2	*len11*	N2	N7	01/09/89	06/06/94
L12	A2	*len12*	N7	N5	01/09/89	06/06/94

Table 6.5 Addition of bypass.

Any change to the location or length of a link would require its replacement in the link table, but not necessarily to its related nodes but the chainage of the end node may well change. Similarly a node will be replaced if it changes junction type which may not necessarily affect its adjoining links. Changes in carriageway width or speed limit that do not affect the location of a link would not require its replacement since this information is held in separate tables. In terms of a GIS each additional table can be thought of as a separate map overlaid on the network table, see figure 6.3.

In the case of speed limit the input of the coordinates of the traffic sign and the speed limit enables the road number, link number and chainage be determined from the previous tables. The intervening links can also be determined from the link table.

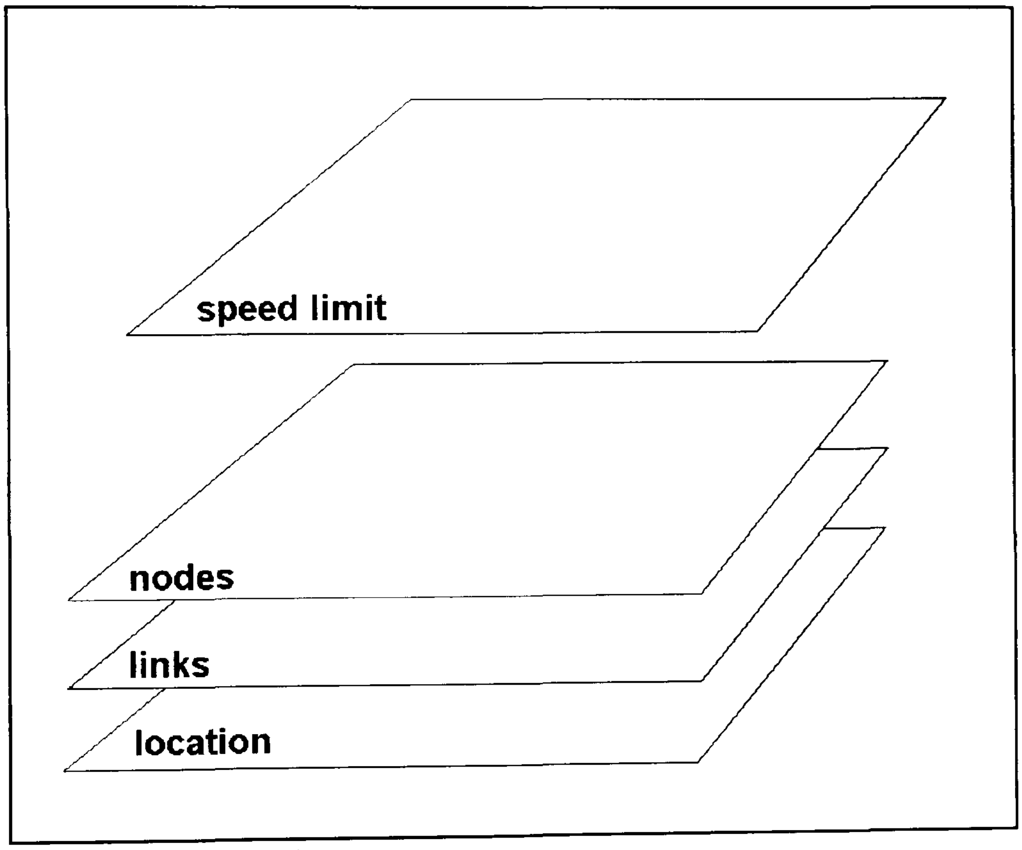


Figure 6.3 Map layers

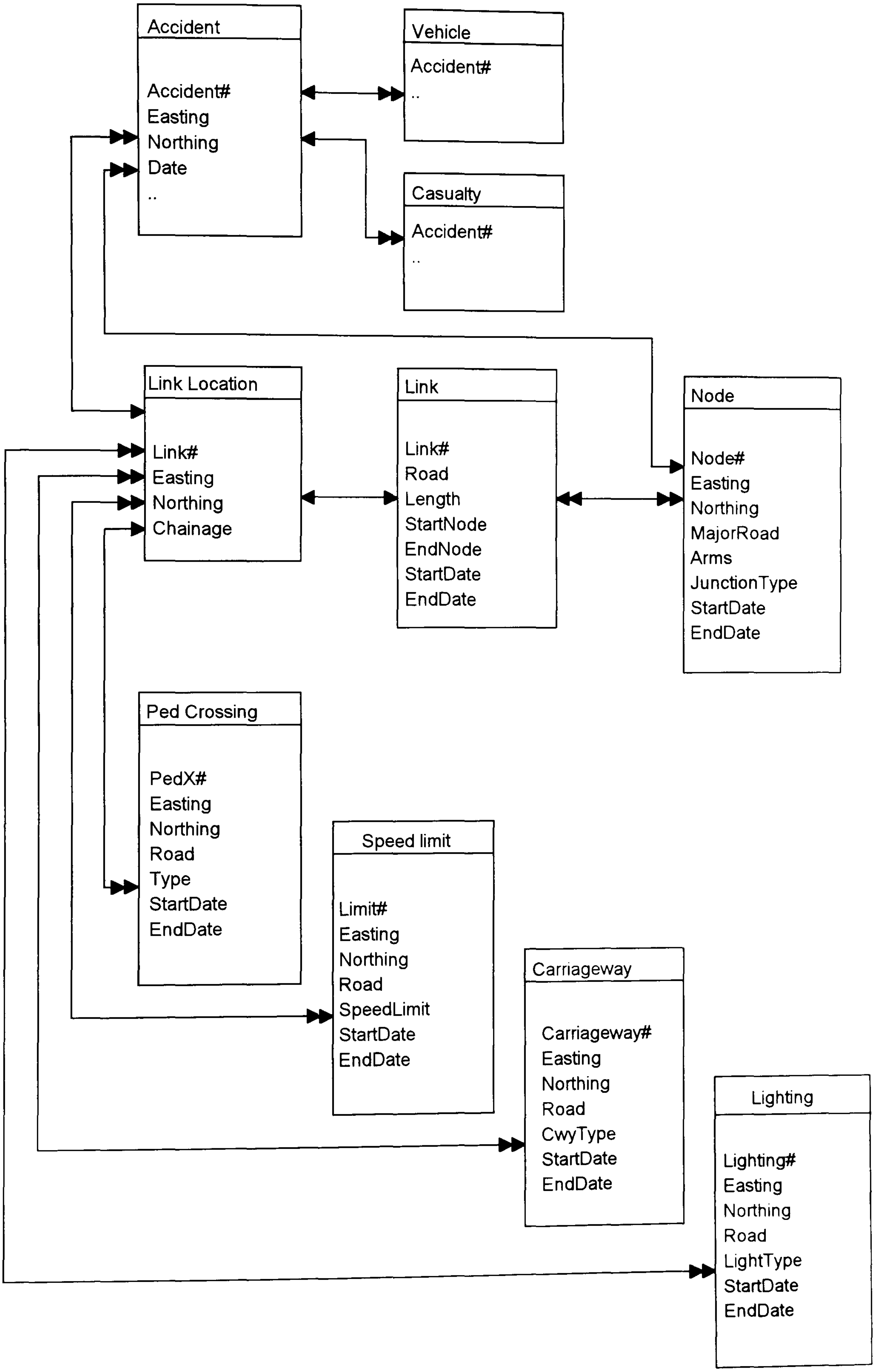


Figure 6.4 Conceptual Schema for an Accident Database

6.4 Conceptual schema for the database

Figure 6.4 shows the conceptual schema for the accident database. Not all of the accident, vehicle or casualty attributes are shown. The degree of the entity relationship is indicated by a single arrow to represent a 'one-to' relationship, and a double arrow for a 'many-to' relationship. The tables show the data to be input by the user. It does not show the attributes or tables which can be determined from these tables, for example, the table showing the connectivity of the network. Other tables may be added to the database, for example, traffic flows or maintenance records, provided suitable relationships are set up by the user. Junctions require further consideration and they are discussed in the next section.

6.5 Junctions

Regarding junctions as points on the network becomes inappropriate as the size of the junction increases, especially when they are grade-separated. As digitised maps become available at larger scale it becomes possible to select those parts of the network where a higher definition is required and to store them in an additional layer, which in this case would be the junction layer. Figure 6.5 shows a grade separated roundabout. The underlying network node is $N1$ which connects links $L1$, $L2$, $L3$ and $L4$, and the drawing illustrates the unsuitability of considering the junction as the point $N1$. In the junction layer the junction now becomes an object $j1$ related to the node $N1$. The junction object is made up of member links $j1.1$ to $j1.9$ that may be connected to, but are not part of the underlying network. This dual definition of the junction enables the accidents at this junction to be compared to other similar junctions by regarding it as a point, or alternatively to be examined in more detail within the members of $j1$.

In the database this junction would be defined by an additional set of network tables linked to the node table. The structure of these new network tables would be the same as the network tables shown in figure 6.4 with each entry and exit being regarded as a separate junction.

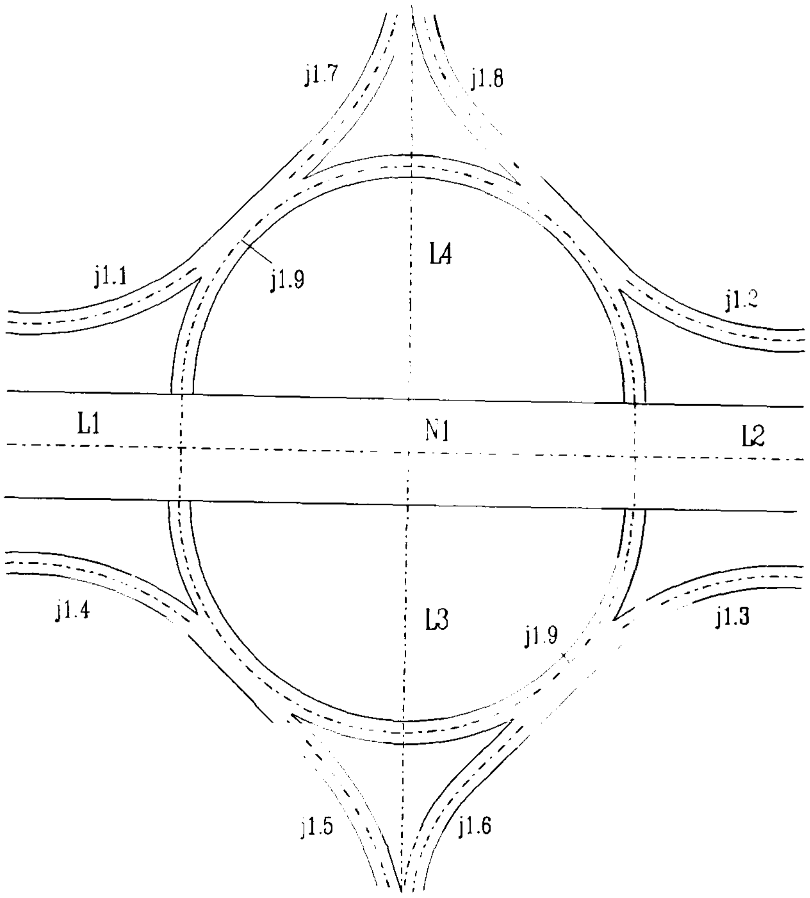


Figure 6.5 Junction Layout

6.6 Minor Junctions

Junctions are generally classified as major or minor. A major junction is a junction between two or more roads of M, A or B classification. For a minor junction the secondary roads are of C classification or lower. Minor junctions are less likely to have a significant effect on traffic distribution and will also tend to experience fewer accidents and are often not specifically defined in the network database.

The speed flow relationships in COBA incorporate a variable which relates to the number of minor intersections and private drives along a road, see Chapter 2. In terms of accidents, minor junctions are not considered separately and are included in the accident rates for links. From a digitised road network it is possible to locate the minor junctions more precisely.

Using Autocad points were snapped on to each minor junction and the coordinates of those points were extracted from a dxf file. The chainage of each junction was then calculated by vector geometry using the accident location program ACCLOC.PRG. An allowance of 20 metres was made either side of this chainage in order to capture the accidents attributed to the junction. Where two junctions were less than 40 metres apart the mid point between the junctions was taken as the boundary. Where two junctions were in close proximity on opposite sides of the major road it was necessary to decide whether these were two separate junctions or a staggered crossroad. This decision was made by inspecting the plan and referring to the junction type in the accident table, but generally if the two roads were clearly a continuation of the same road then the junction was taken to be a staggered crossroad.

It could be argued that in the situation where two side roads are directly opposite each other but not part of the same road that they should nevertheless be considered to be a crossroad. The alternative would be to consider them as two separate tee junctions at the same location but with different secondary road numbers, but this would imply that crossing movements never occur, even though they may be comparatively rare.

A minor junction will generally have a much lower flow on the secondary road than at a major junction and it can be argued that since minor junctions tend to be less well defined from the drivers point of view it can be the rarity of the turning movements at minor junctions that create a hazard. Another consideration for minor junctions is that accidents are still likely to occur on the major road when turning movements are not a factor. For this reason an accident was only attributed to a minor junction if it was recorded as a junction accident. After the junctions were located a manual check was carried out to ensure that the accidents were correctly attributed to the relevant junction.

For the network in Kent the numbers of junctions and accidents are shown in table 6.6. The total number of accidents on the network from 01/01/84 to 31/12/91 was 5479. 36.2% of all accidents were attributed to major and minor junctions, not including private driveways, and 24.9% were attributed to minor junctions.

Junction Category	jncs	accs	% total
Major	50	620	11.3
Minor	627	1364	26.9
Total for jncs	677	1984	36.2

Table 6.6 Accidents at junctions.

6.7 Analysis of Accidents at minor junctions

Figure 6.6 shows the frequency distribution for accidents at minor junctions over a five year period from 1984 to 1988. It is usually assumed that the frequency distribution for accidents over a number of similar sites is of the negative binomial form [19]. A negative binomial distribution was fitted to the data using the statistical package MLP [20]. For the distribution $X^2 = 14.94$ with 21 degrees of freedom compared to X^2 $= 32.67$ at the 5% level of significance. These results show a very good fit to the negative binomial distribution. This result indicates that it is possible, and indeed desirable, to analyse accidents at minor junctions separately. At present the accident rates for links in COBA include both link and minor junction accidents which is

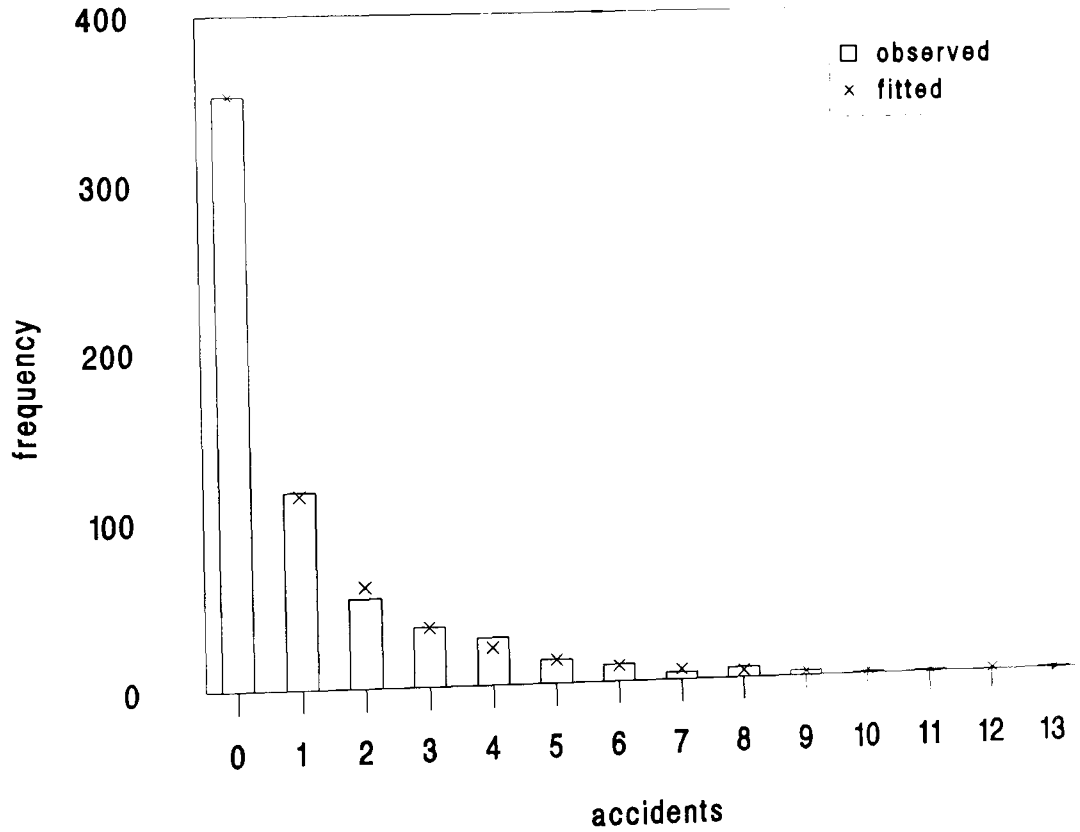


Figure 6.6 Negative binomial distribution of accidents at minor junctions

inadequate for the analysis of accident sites.

Local Authorities carry out improvements to the highway that are designed to reduce the number of accidents. In order to assess the effectiveness of these measures it is necessary to compare the number of accidents after the treatment with a prediction of the number that would have occurred if no improvement had taken place. The best estimate of the number of accidents y in after period is the underlying mean number of accidents m for that site. Initially m is unknown, but can be estimated using an Empirical Bayes method [19]. In this method a distribution is first assumed for the means at a number of sites which is updated as more information becomes available about the sites. The additional information consists of the observed number of accidents at those sites. It can be shown that if the observed accident frequencies follow a negative binomial distribution then the means will follow a gamma distribution and that the mean $\hat{m}$ of this distribution can be calculated from the equation

$$\hat{m} = \frac{(k+x)}{(\alpha+1)}$$

where x = number of accidents in the before period and k and α are parameters of the negative binomial distribution.

The validity of this method was checked by comparing accidents at minor junctions for 3 year before and after periods. A negative binomial distribution was fitted to the accidents in the before period for which $k = 0.3873$ and $\alpha = 0.5484$. Table 6.7 shows the accident frequency and $\hat{m}$ calculated from the formula above. The average number of accidents which occurred at those sites in the after period $\bar{y}$ is also shown.

One method of assessing $\hat{m}$ as an efficient predictor of $\bar{y}$ if their difference is less than two standard errors, [21] which is clearly the case as indicated in the righthand column. The standard error may be calculated from the formula
where f = the number of sites at which x accidents were observed in the before period.

$$SE = \sqrt{\frac{(\hat{\alpha}+2)(\hat{k}+x)}{f(1+\hat{\alpha})^2}}$$

x	f	$\hat{m}$	$\bar{y}$	$(\hat{m}-\bar{y})/SE$
0	367	0.2501	0.2779	-0.8305
1	89	0.8960	0.8090	0.6757
2	44	1.5418	1.4773	0.2687
3	21	2.1876	1.7619	1.0282
4	11	2.8335	1.8182	1.5593
5	6	3.4793	2.3333	1.1731
6	5	4.1252	3.4000	0.6223
7	4	4.7710	4.0000	0.5503
15	1	9.9378	11.0000	-0.2627

Table 6.7 Predicted and calculated means

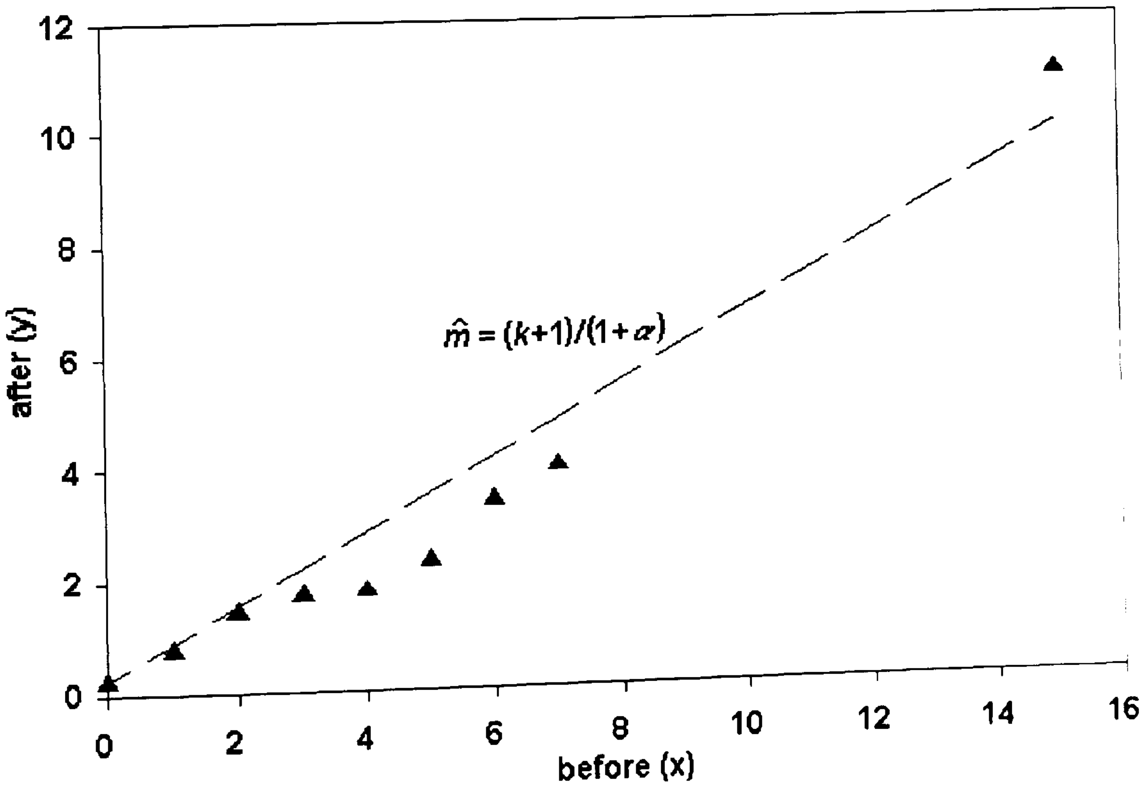


Figure 6.7 $\hat{m}$ as a predictor of accidents in the after period

Figure 6.7 shows a graphical representation of the results. The accidents in the after period are plotted against the accidents in the before period and the linear relationship between the mean and the before accidents is also shown. The data appears to be exhibiting an upward trend which is steeper than the calculated values of $\hat{m}$. However, this is due to the influence of one site where x is equal to 15 and so does not constitute firm evidence that the calculated relationship is inappropriate. The effectiveness of these results show that it is possible to define the road network in greater detail without losing the ability to carry out analysis of those sites. In the case of minor junctions the fact that they tend to have fewer accidents than major junctions is more than offset by their greater frequency. Similar analysis could possibly be applied to pedestrian crossings and even, perhaps, pedestrian refuges if they were defined in the database.

6.8 Conclusion

Defining the road network separately in the database means that there is scope for defining it in greater detail, enabling a greater understanding of the relationship between the number of accidents associated with elements of the road network.

In this chapter a method has been developed for incorporating the definition of the road network into the accident database. This method is based on the link and node system currently used in programs for traffic analysis and which is suited to a computerised database. It also possible to provide information not currently available such as the number of accidents at minor junctions.

However, the definition of the road network at various levels of detail and the incorporation of procedures for accident analysis tend to suggest that an object-oriented system is more appropriate than a relational database system. The principles and the applicability of an object-oriented system are considered in the next chapter.

Chapter 7
Object-Oriented Databases

7.1 Introduction

The previous chapters have indicated that there is a need to develop a hierarchical representation of the road network, for example, we may need to consider accidents on an entire road, or on individual links, or the components which make up a link such as minor junctions or sharp bends. If it is also necessary to calculate accident rates for these entities within the database then this suggests that an object-oriented approach is more appropriate.

In many local authority accident databases the road network has been envisaged as an attribute of accident. This does not coincide with a natural perception of the data. The highway engineer often wishes to ask questions from a perspective, such as "Are there too many accidents at this site and if so, why?" The entity has now becomes the site which possesses the attribute accident.

Object-oriented databases have recently developed from the concepts of object-oriented programming languages. In an object-oriented approach the engineer's concept is stated as a formal procedure to be incorporated into the database, this concept is illustrated in the following quotation:

> To manipulate data and produce information, a scientist needs to access data and apply analysis tools in concert. Failure to integrate the data management and analysis environment restricts the productivity of the scientist.... [22]

7.2 Object-Oriented Database Systems [23]

Object-oriented programming is an attempt to model objects as we perceive them in the real world. In our vocabulary we have invented nouns which describe a broad classification of objects, for example, person, car or building. This enables us to distinguish among groups of objects without being distracted by the particular details of objects, for example, the word aeroplane represents a class of object which is quite different to a car. From the object class aeroplane we can develop sub-classes of aeroplane which have their own particular characteristics; for example airliners, executive jets, single-seaters and so on.

In a computer program a function is usually available to any data of relevant type within the scope of the program. This may not always be desirable, for example, if a function calculates the expected number of accidents at a junction there is little point in this function being available to non-junction data. The binding together of data and the functions which operate on that data into objects is a characteristic of object-oriented programming known as *encapsulation*.

As an example, in a GIS the road network can be represented by combinations of points and arcs. A point can represent the location of a node or an accident, an arc a section of a link between two digitised points. The definition of the object classes Point and Arc are as follows:

```
class Point
{
        Easting
        Northing
}

class Arc
{
        ArcID#
        StartPoint
        EndPoint
        Length()
}
```

The function **Length()** is only available to the coordinates of **Startpoint** and **Endpoint** and will reject any other data. In this context the database can be said to possess a degree of intelligence since it is able to recognise data which is appropriate for the function. The attributes of **Point** and **Arc**, are themselves objects and so an object type is defined in terms of other object types. In this context an object is regarded as a collection or *aggregation* of its attributes and is equivalent to a row in a table.

Each object is associated with a unique identifier, independent of the data, which is determined by the system. By contrast, in the relational system objects, or records, are uniquely identified by a combination of attributes determined by the user. The object-oriented database therefore provides greater data integrity than the relational database and search operations are also more efficient.

A class of objects can be defined from which other *sub-classes* can be derived, the property of *inheritance*. The sub-classes inherit the attributes and functions of the original or *super-class*. For example, a class **Junction** has sub-classes **Roundabout, TrafficSignal** and **MajorMinor**.

A further characteristic is *polymorphism*. An function bearing the same name may be shared by different objects but take on a different form for each object. For example, the function which calculates expected number of accidents may vary according to junction type.

The object-oriented database allows a more complex representation of data. In a relational database each cell is *atomic,* it can only contain a single value, in the object-oriented database a 'cell' could contain a photograph and the functions to display that photograph on the screen. Whilst visual displays can be incorporated into a relational database it can only be achieved via external routines which are less efficient.

SQL, the query language for relational databases, does not allow highly complex queries and may not blend with the syntactic structure of the host language. Query languages for object-oriented databases tend to be written in the host language, or a sub-set of that language, which allows the developer to provide more complex queries but also provide interactive queries for the user.

7.3 Application to the Road Network

When relating to the road network we usually refer to the number or name of a particular road, for example, the A2 or the High street. The road name may serve as a reference to a particular location, such as 17 High street, or it may identify a route, the A2 from London to Dover. The same road may represent a set of routes, for example the A2 is also the route from Rochester to Canterbury, it may also form part of another route, the A2 and the A299 from London to Margate. It is usual therefore to think of a road as a series of independent links connected by junctions.

The basic identity of a road consists of its name and start and end point which remains unchanged despite physical changes to the road itself. The A2 has always been the London to Dover road but its precise route has been changed by the addition of a number of bypasses. It was once all single carriageway, but now it is partly dual carriageway. The physical characteristics of a road are attributes of the road and not part of its basic identity. In the object-oriented world **road** is the super-class for the road network and comprises a name and a start and end point.

A distinction should be made between a node and a junction on the road network. A node generally refers to a point on the network, either an intersection, or a boundary. An intersection is not necessarily a junction, it could be a fly-over, and a boundary could refer to a local authority boundary or a change in speed limit. A junction, however facilitates movement from one link to another and is also a physical object, often of some size, such as a grade separated junction. A set of links and junctions make up the road. We think of a junction as a junction between particular roads and a link is a section of road joining two junctions.

7.4 Stats 19 data

A similar approach can be adopted to accident data which would include vehicle objects and casualty objects. All casualties have three attributes in common, age, sex and severity; these are the attributes of the object class **casualty**. The age of the casualty will determine whether the casualty is also a member of the class **schoolchild**. The class casualty can have two sub-classes **pedestrian** and **vehicle occupant**. The latter being an attribute of the relevant **vehicle** object. The type of vehicle will determine further attributes, for example, if the vehicle is a car then the attribute seat belt usage will apply.

This approach avoids the anomalies possible using a relational system where it is possible to attribute seat belt usage to a pedestrian.

7.5 Conclusion

There is scope for improvement in the way that local authority accident data is organised. Relational databases are not ideally suited to more complex relationships and calculations, particularly geometric calculations. Object-oriented databases are more suited to this approach and there is a need to investigate the use of this type of database for storing accident data.

Chapter 8
Conclusion

There is considerable scope for the development of road accident databases. One major disadvantage of present systems is that the road network is not adequately defined. There are a variety of methods used by local authorities for representing the road network and for locating accidents. They do not, however, provide a separate definition of the network which means that an element of the network does not exist within the database until an accident has occurred on it. The data required for the statistical analysis of accidents is therefore incomplete and there is a tendency to bias the selection of sites toward those that experience the highest number of accidents.

The present systems were designed before the advent of digital maps and geographical information systems. In the past, it may have been considered impracticable to provide a separate network definition when the data required extraction from paper maps and by site survey. Some of the parameters used to describe the road geometry, such as *bendiness*, were conceived assuming manual methods of extraction.

Digitised maps contain a detailed description of the road centre line and the location of junctions, including minor junctions. During this study methods have been investigated for representing the centre line by a continuous alignment, using cubic splines, and to provide an estimate of curve radius. It was found that, for the available data, the regression spline technique was the most successful. Methods were then investigated for the relationship between the expected number of accidents and curve radius and traffic flow. Whilst the resulting models were not entirely satisfactory, there is scope for their development by including additional variables, such as curve length.

After the centre line was defined, methods were developed for representing the network using a relational database and the basis for a conceptual schema is proposed. An important aspect of the database is that it should allow the representation of the network at different levels of detail and the inclusion of changes to the road network.

The locations of minor junctions are available from digital maps. These locations were extracted and the frequency distribution of accidents was examined. This resulted in a near perfect fit with a negative binomial distribution, which may be due to the fact that there were an unusually large number of similar sites available for analysis. The availability of this data means that it is possible to separate the accidents at minor junctions from other accidents on a link site and provide more accurate analysis.

The hierarchical nature of the road network definition combined with the calculation of road geometry and expected accident frequencies suggest that an object-oriented database system, with its hierarchical structure and facility for performing complex calculations, is more appropriate than a relational database system. The concepts of object-orientation has been discussed and is proposed as a direction for further research.

The analysis of the factors that influence road accidents has outstripped the availability of adequate data. This thesis has suggested improvements to accident database systems by using new database technology and by organising existing data in a way that enables its potential to be fully realised.

References

[1] A.J. Boyle and C.C. Wright, "Accident 'migration' after remedial treatment at accident blackspots". Traffic Engng & Control 25 (5). (1984), pp 260-267.

[2] M.Maher, "Accident Migration - a statistical explanation?", Traffic Engng & Control 28 (9). (1987), pp 480-483.

[3] L. Mountain and B. Fawaz,"The area-wide effects of engineering measures on road accident occurence". Traffic Engng & Control 30 (7/8). (1989), pp 355-360.

[4] D.R. Fraser Taylor, "Geographic Information Systems", Pergamon Press, Oxford. (1991).

[5] PC Arc/Info, White Paper Series, Environmental Systems Research Institute Inc. (1992).

[6] A. Peled and A. Shalom Hakkert, "A PC-oriented GIS application for road safety analysis and management", Traffic Engng & Control 34 (7/8). (1993), pp 355-361.

[7] Department of Transport, COBA 9 Manual. (1989).

[8] D.R. Howe, "Data analysis for data base design", Edward Arnold, London. (1983).

[9] K. Austin, "The identification of mistakes in road accident records", Accident Analysis & Prevention, Vol 27, No 2, pp 261-276, 277-282.

[10] C. De Boor, "Bicubic Spline Interpolation", J. Math. Phys. 41. (1962), pp. 212-218.

[11] C. De Boor, "A Practical Guide to Splines", Springer-Verlag, Berlin. (1978).

[12] P. Lancaster and K. Salkauskas, "Curve and Surface Fitting, An Introduction", Academic Press, London. (1986).

[13] T.N.T. Goodman and K. Unsworth, "Shape Preserving Interpolation by Parametrically Defined Curves", SIAM J. Numer. Anal. 25 (1988), pp. 1-13.

[14] R. Delbourgo, "Accurate C^2 Rational Interpolants in Tension", SIAM J. Numer. Anal. 30. (1993), pp. 595-607.

[15] S. Miaou and H. Lum, "Modelling vehicle accidents and highway geometric design relationships". Accident Analysis & Prevention, Vol. 25, No 6. (1993) pp 689-709.

[16] G. Maycocok and R.D. Hall, "Accidents at 4-arm roundabouts", Department of Transport TRRL Report 1120, Transport Research Laboratory, Crowthorne.

[17] R.J. Baker and J.A. Nelder, "Glim Manual", Numerical Algorithms Group, Oxford. (1978).

[18] A.J. Dobson, "Introduction to Statistical Modelling", Chapman and Hall, London. (1983).

[19] C. Abbess, D.F. Jarrett & C.C. Wright, "Accidents at blackspots: Estimating the effectiveness of remedial treatment, with special reference to the 'regression-to-mean' effect", Traffic Engineering & Control 22 (10). (1982), pp 535-542.

[20] G.J.S. Ross, "MLP, Maximum Likelihood Program", Numerical Algorithms Group. (1980).

[21] D.F. Jarret, C. Abbess & C.C. Wright, "Bayesian Methods applied to recent road accident blackspot studies: some recent progress", Seminar on Short-term and Area-wide Evaluation of Safety Measures, Institute for Road Safety Research SWOV, Amsterdam. (1982).

[22] J.C. French, A.K. Jones and J.L. Pfaltz, "Scientific database management ", Department of CS, University of Virginia. (1990).

[23] M.Chignell, S. Khoshafian, K. Parsaye, H. Wong, "Intelligent Databases", John Wiley & Sons, New York. (1989).

Appendix A

DATA REQUIRED IN STATS19 CODING FORM.

ATTENDANT CIRCUMSTANCES RECORD

Accident Reference Number
Severity Fatal, severe, slight
No. Vehicles
No. Casualties
Date **Day of week**
Time
1st Road No. **2nd Road No.**
Speed limit
Verbal description of location
Verbal description of accident
Grid reference

ROAD TYPE

1 Roundabout
2 One way street
3 Dual c'way 2 lanes
4 Dual c'way 3 or more lanes
5 Single c'way single track
6 Single c'way 2 lanes
7 Single c'way 3 lanes
8 Single c'way 4 or more lanes

JUNCTION

0 Not within 20 metres
1 Roundabout
2 Mini roundabout
3 'T' or staggered
4 'Y'
5 Slip road
6 Cross roads
7 Multiple
8 Private drive/entrance
9 Other

JUNCTION CONTROL

1 Authorised person
2 Automatic traffic signal
3 Stop sign
4 Give way sign
5 Uncontrolled

PEDESTRIAN CROSSING

0 None within 60 metres
1 Zebra
2 Zebra, Sch crossing
3 Zebra, Other Auth person
4 Pelican
5 Other, light controlled
6 Other, Sch crossing
7 Other, Auth Person
8 Central refuge, no control
9 Footbridge or subway

LIGHT CONDITIONS

Footway lighting <7m; Road lighting >7m.

LIGHT

1	Over 7m high
2	Under 7m
3	none
4	Not known

DARK

5	Over 7m high
6	Under 7m
7	No street lighting
8	Street lights unlit
9	Not known

WEATHER

1	Fine
2	Rain
3	Snow
4	Fine (High winds)
5	Rain (High winds)
6	Snow (high winds)
7	Fog (or mist)
8	Other
9	Unknown

ROAD SURFACE

1	Dry
2	Wet/damp
3	Snow
4	Frost/ice
5	Flood

CARRIAGEWAY HAZARDS

1	None
2	Dislodged load
3	Other object
3	Previous Accident
4	Dog
5	Other animal

SPECIAL CONDITIONS

0	None
1	ATS out
2	ATS defective
3	Previous Accident
4	Road works
5	Surface defective

VEHICLE RECORD

TYPE OF VEHICLE

01	Pedal cycle
02	Moped
03	Motor scooter
04	Motor cycle
05	Combination
06	Invalid tricycle
07	Other 3 wheeled car
08	Taxi
09	Car four wheeled
10	Minibus/motor caravan
11	PSV
12	Goods not over 1.5 tons
13	Goods over 1.5 tons
15	Other non-motor vehicle

TOWING & ARTICULATION

0	Not towing
1	Articulated vehicle
2	Double/multiple trailer
3	Trailer caravan
4	Single trailer
5	Other tow

MANOEUVRES

01	Reversing
02	Parked
03	Held up
04	Stopping
05	Starting
06	U-turn
07	Turning left
08	Waiting to turn left
09	Turning right
10	Waiting to turn right
11	Changing lane to left
12	Changing lane to right
13	Overtaking offside moving veh
14	Overtaking offside stationary veh
15	Overtaking on nearside
16	Going ahead LH bend
17	Going ahead RH bend
18	Going ahead other

VEHICLE LOCATION

01	Leaving main road
02	Entering main road
03	On the main road
04	On minor road
05	On service road
06	On lay-by/hard shoulder
07	Entering lay-by/hard shoulder
08	Leaving lay-by/hard shoulder
09	On cycleway
10	Not on carriageway

VEHICLE JUNCTION LOCATION AT FIRST IMPACT

0	Not at or within 20 metres
1	Approaching junction/parked at junction approach
2	Vehicle in middle of junction
3	Clear of junction/parked at exit
4	Did not impact

SKIDDING

0	Not applicable
1	Skidding
2	Skidded and overturned
3	Jack-knifed
4	Jack-knifed and overturned
5	Overturned

FIRST POINT OF IMPACT

0	Did not impact
1	Front
2	Back
3	Offside
4	Nearside

VEHICLE LEAVING CARRIAGEWAY

0	Did not leave carriageway
1	Left c'way nearside
2	Left c'way nearside and rebounded
3	Left c'way straight ahead at junction
4	Left c'way offside onto central reservation
5	Left c'way offside onto central reservation and rebounded
6	Left c'way offside and crossed central reservation
7	Left carriageway offside
8	Left carriageway offside and rebounded

DAMAGE

0	None	4	Nearside
1	Front	5	Roof
2	Back	6	Underside
3	Offside	7	All 4 sides

OBJECT IN CARRIAGEWAY HIT

00	No object hit	06	Bridge (side)
01	Previous accident	07	Bollard/Bridge
02	Roadworks	08	Open door of vehicle
03	Parked vehicle lit	09	Central island of roundabout
04	Parked vehicle unlit	10	Kerb
05	Bridge (roof)	11	Other object

OBJECT OFF CARRIAGEWAY HIT

00	No object hit	06	Central crash barrier
01	Road sign/signals	07	Nearside/offside crash barrier
02	Lamp post	08	Submerged
03	Telegraph pole	09	Entered ditch
04	Tree	10	Other permanent object
05	Bus stop/shelter		

OTHER VEHICLE HIT

Vehicle reference no.

BREATH TEST

0	Not applicable	3	Not requested
1	Positive	4	Failed to provide
2	Negative	5	Driver not contacted at time

DIRECTION OF TRAVEL INCLUDING COMPASS POINT

From and to.

CASUALTY RECORD

CASUALTY CLASS

1	Driver or rider	3	Pedestrian
2	Passenger		

SEX 1 Male, 2 Female

Age

Severity

PEDESTRIAN LOCATION

00	Not a pedestrian
01	On pedestrian crossing
02	Within zigzag lines on approach to crossing
03	Within zigzag on exit to crossing
04	In c'way elsewhere within 50 m of crossing
05	In carriageway crossing elsewhere

06	On footway, verge
07	On refuge or central reservation or island
08	In centre of c'way not on reservation or refuge
09	In carriageway nor crossing
10	Unknown

PEDESTRIAN MOVEMENT

0	Not a pedestrian
1	Crossing from driver's nearside
2	Crossing from nearside, masked by stationary vehicle
3	Crossing from driver's offside
4	Crossing from driver's offside, masked by stationary vehicle
5	In carriageway - not crossing
6	In carriageway - not crossing masked by stationary vehicle
7	In carriageway, walking, facing traffic
8	In carriageway, walking, back to traffic
9	Unknown or not in carriageway

SCHOOL PUPIL

0	Not a pupil
1	Travelling to or from school
2	Not travelling to or from school

SAFETY BELT USAGE

0	Casualty not a car occupant
1	In use
2	Fitted not in use
3	Not fitted
4	Child harness in use
5	Child harness not in use
6	Child harness not fitted
7	Not known

PSV PASSENGER

0	Casualty not on PSV
1	Boarding
2	Alighting
3	Standing
4	Seated

CAR PASSENGER

0	Not a car passenger
2	Front seat passenger
3	Rear seat passenger

PEDESTRIAN MOVEMENT

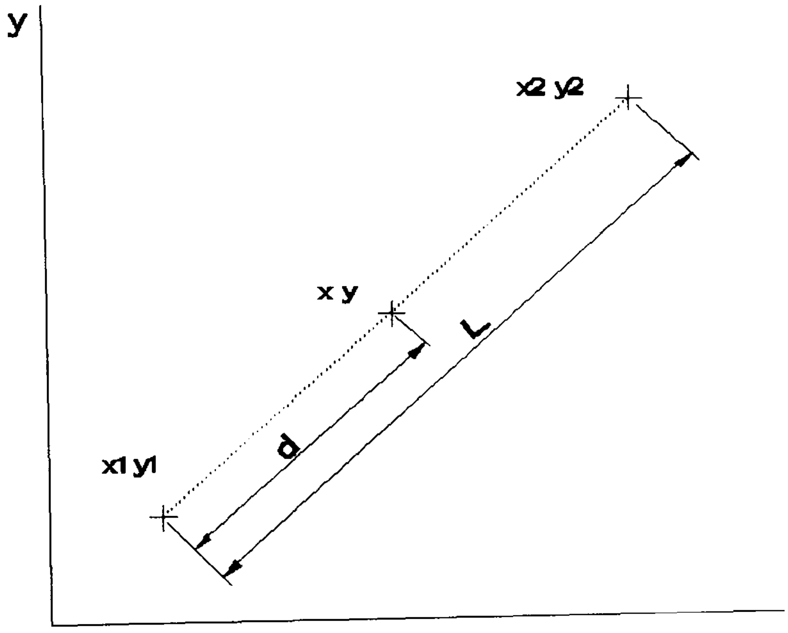
From and to

Appendix B

VECTOR GRAPHICS

This appendix describes the techniques of vector geometry used in chapter 4.

Equation of a straight line



Consider a line fixed by two points

$$p_1 \equiv (x_1y_1)\ p_2 \equiv (x_2y_2)\ ,$$

and a general point on the line

$$p \equiv (xy).$$

Length of the line p_1 to p_2 = L and p_1 to p = d .

The coordinates represented by p are determined by

$$x = x_1 + \frac{d(x_2 - x_1)}{L},$$

let $\mu = d/L$ then

$$x = x_1(1-\mu) + \mu x_2$$

similarly

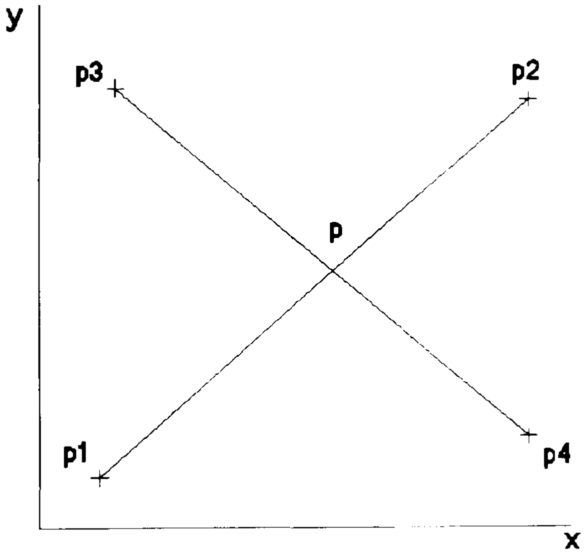
$$y = y_1(1-\mu) + \mu y_2$$

which gives the general equation for x,y

$$p(xy) = (1-\mu)p_i + \mu p_{i+1}$$

For p to lie between p_1 and p_2 then $0 \leq \mu \leq 1$

Intersection of two lines



Equation of a point on line 1 (p_1,p_2)

$$(1-\mu)p_1 + \mu p_2$$

and on line 2 (p_3,p_4)

$$(1-\sigma)p_3 + \sigma p_4$$

point p, the intersection lies on both lines so

$$(1-\mu)x_1 + \mu x_2 = (1-\sigma)x_3 + \sigma x_4$$

$$(1-\mu)y_1 + \mu y_2 = (1-\sigma)y_3 + \sigma y_4$$

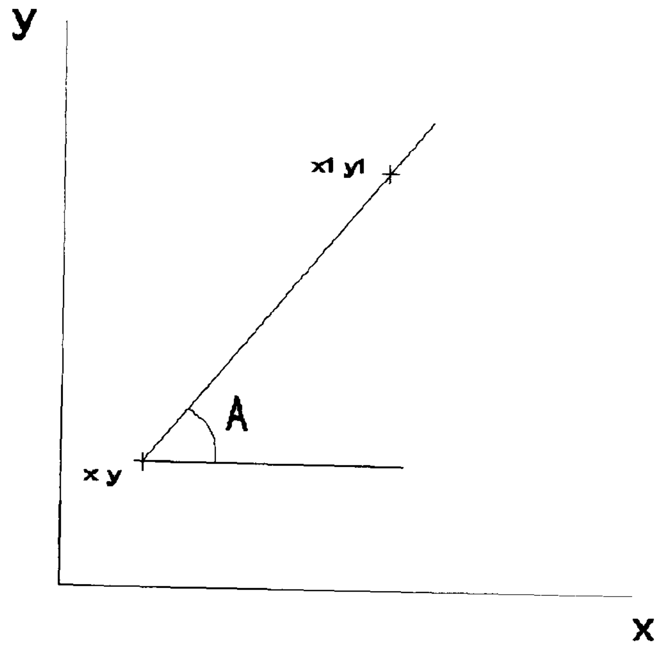
solving

$$\mu = \frac{(x_3-x_1)(y_3-y_4)-(x_3-x_4)(y_3-y_1)}{(x_2-x_1)(y_3-y_4)-(x_3-x_4)(y_2-y_1)}$$

p can be calculated from

$$p \equiv (1-\mu)p_1 + \mu p_2$$

Direction Cosines



Consider a point xy on a line which makes an angle *A* with the x axis then the directional cosines are

$$c \equiv (c_x c_y)$$

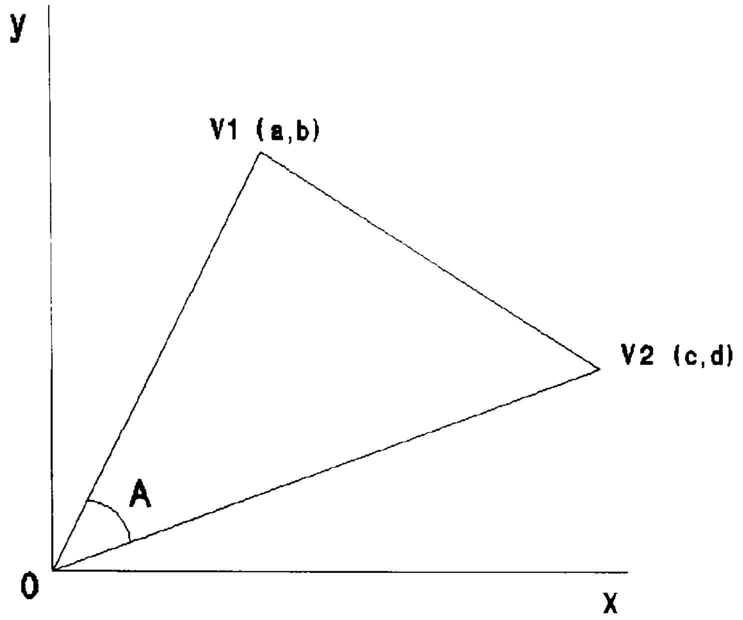
where $c_x = cos\mathrm{A}$ and $c_y = cos(90\text{-A})$ and so

$$c_x = \frac{x}{\sqrt{(x^2+y^2)}}, c_y = \frac{y}{\sqrt{(x^2+y^2)}}.$$

A line joining two points p_1 and p_2 has directional cosines

$$\frac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}, \frac{(y_2-y_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}$$

Angle between two lines having directional cosines ab and cd



The distance from the origin to each point

$$OV_1 = OV_2 = 1$$

by the cosine rule

$$V_1V_2^2 = OV_1^2 + OV_2^2 - 2.OV_1.OV_2.cos\mathrm{A} = 2(1-cos\mathrm{A})$$

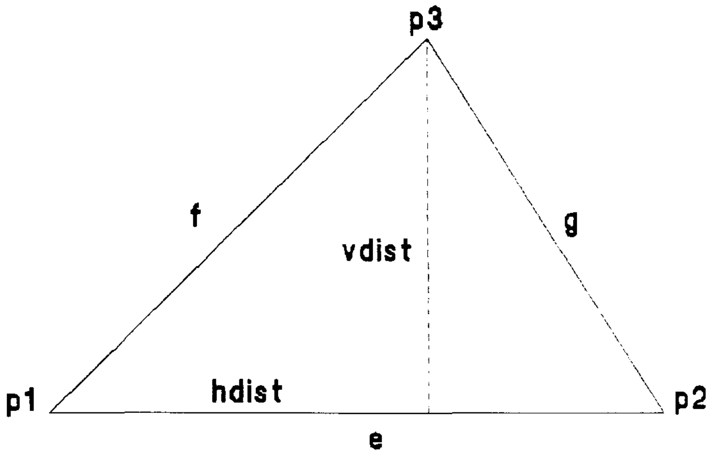
also by Pythagoras

$$V_1V_2^2 = (a-c)^2 + (b-d)^2 = 2 - 2(ac+bd)$$

after equating these

$$cos\mathrm{A} = (ac+bd)$$

Perpendicular offset from a point to a line



Consider three general points p_1,p_2 and p_3

$$a=\frac{(x_2-x_1)}{e},b=\frac{(y_2-y_1)}{e},c=\frac{(x_3-x_1)}{f},d=\frac{(y_3-y_1)}{f}.$$

then

$$\cos A=\frac{(x_2-x_1)(x_3-x_1)+(y_2-x_1)(y_3-y_1)}{ef}=\frac{ac+bd}{ef}$$

and so

$$hdist=f\cos A=\frac{(ac+bd)}{e}$$

$$vdist=\sqrt{(f^2-hdist^2)}$$

Appendix C

```
* FOXPRO2 PROGRAM CHAIN.PRG

* Calculates chainage.
* Requires input from a dbf file in the form
* {road number, easting, northing, chainage}

* K. Lupton 1993

close all
clear

dbf=getfile('dbf','Open Database')
        use (dbf)

*       x = horizontal distance between points
*       y = vertical distance between points
*       z = distance between points
*       c = chainage of points in km

c=0
select  from (dbf) into array users

for i=1 to alen(users,1)-1
        x=users(i+1,2)-users(i,2)
        y=users(i+1,3)-users(i,3)
        z=sqrt(x^2+y^2)/1000
        c=c+z
        if users(i+1,1)<>users(i,1)       && if change in road number,
                c=0                       && set chainage to zero.
        endif
        users(i+1,4)= z
endfor

delete all
pack
append from array users
```

```
* FOXPRO2 PROGRAM ACCLOC.PRG

* Calculates chainage of accidents on road alignment.
* Accidents must be in specified zone parallel to centreline or
*  within specified radius from node.

* K. Lupton 1993

        net.dbf  = {rnd, easting, northing, chainage}
        acc.dbf =  {accid, rnd, easting, northing, chainage}

clear
set talk off
close all

? "Select network file"
dbf2 = getfile('dbf','Open Database')
use (dbf2)
copy to array net

?"Select accident file"
dbf1 = getfile('dbf','Open Database')
use (dbf1)
copy to array acc

jn = alen(net,1)                                && number of nodes
ja = alen(acc,1)                                && number of accidents

zone = 20
zone_sq = zone^2
radius = 20
rad_sq = radius^2

FOR  i = 1 TO jn-1
        a = net(i+1,2) - net(i,2)               && easting
        b = net(i+1,3) - net(i,3)               && northing
        e = net(i+1,4) - net(i,4)               && link length

        FOR k = 1 TO ja
                IF acc(k,5) = 0
                        c = acc(k,3)-net(i,2)
                        d = acc(k,4)-net(i,3)
                        hdist = (ac+bd)/e
                        IF hdist > = 0 and hdist < e
                                f_sq = c^2 + d^2     && dist from node to acc
                                IF f_sq < = rad_sq
                                        f_sq = 0
                                        hdist = 0
                                ENDIF
                                vdist_sq = f_sq-hdist^2
                                IF  vdist_sq < zone_sq
                                        acc(k,5) = (hdist/1000) + net(i,4)
                                ENDIF
                        ENDIF
```

```
            ENDIF
        ENDFOR
ENDFOR

use (dbf1)
delete all
pack
append from array acc



* FOXPRO2 PROGRAM ANGMES.PRG

* Calculates Deflection Angle between successive Data Points

* K. Lupton 1993

    close all
    clear
    set talk off
    dbf = getfile('dbf','Open Database')
    use (dbf)
    copy to array arr
    n = alen(arr,1)

    for i=1 to n-2

        a = (arr(i+1,2)-arr(i,2))
        b = (arr(i+1,3)-arr(i,3))
        c = (arr(i+2,2)-arr(i+1,2))
        d = (arr(i+2,3)-arr(i+1,3))

        dist2 = sqrt((aa+bb)(cc+dd))
        cosan = (ac+bd)/dist2
        arr(i+1,5) = (pi()-acos(cosan))180/pi()

    endfor

    arr(1,5) = 0
    delete all
    pack
    append from array arr
```

CUBIC SPLINES

```
// CUBSCR.CPP
// Turbo C+ +
// K. Lupton. August 1993
// CALCULATION OF EXISTING ROAD ALIGNMENT USING CUBIC SPLINES
// PARAMETERISED BY CURVE LENGTH.
// Produces Autocad script file

#include <fstream.h>
#include <strstrea.h>
#include <string.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>

#define N 500
ofstream fpout;

void Open_datafile(char road[],long x[],long y[],int rec);
void curve_len(long x[],long y[],float ch[],int n);
float spline_calc(float p[],long q[],float s[][4],int flen);
char change_name(char name[],char ext[]);

main()
{
        int i,k,rec=0;
        long x[N+1],y[N+1],cpos=0;
        float sx[N+1][4],sy[N+1][4],ch[N+1];
        float xpos,ypos,w;
        char road[12];
        char scr[]=".scr";

        x[0]=y[0]=sx[0][0]=sy[0][0]=ch[0]=0;

        Open_datafile(road,x,y,&rec);
        curve_len(x,y,ch,rec);
        spline_calc(ch,x,sx,rec);
        spline_calc(ch,y,sy,rec);
        change_name(road,scr);
        fpout.open(road,ios::out);

// Calculate spline coordinates //
        k=0;
        fpout< <"PLINE ";
        do
        {
                for(i=1;i< =rec;i+ +)
                {
                        if(cpos< =ch[i])
                                {  k=i-1;break;}
                }
                        w=cpos-ch[k];
                        xpos=((sx[k][3]w  +  sx[k][2])w  +  sx[k][1])w
```

```
                                    + sx[k][0];
                        ypos=((sy[k][3]w + sy[k][2])w + sy[k][1])w
                                    + sy[k][0];
                        fpout< <xpos< <","< <ypos< <'\n';
                        cpos+ =10;
          } while(cpos< =long(ch[rec])) ;
          fpout< <x[rec]< <","< <y[rec];
          fpout.close();
          return 0;
}

void curve_len(long x[],long y[],float ch[],int n)
{
          int i;

          for(i=1;i<n+1;i+ +)
          {
                    ch[i]=ch[i-1]+hypot(x[i]-x[i-1],y[i]-y[i-1]);
          }
          return;
}

float spline_calc(float p[],long q[],float s[][4],int flen)
{
          int i;
          float h[N],d[N],a[N],b[N],c[N],v[N],m[N+1];
          float f;
          h[0]=d[0]=a[0]=b[0]=c[0]=v[0]=0;

          h[0]=p[1]-p[0];
          d[0]=(q[1]-q[0])/h[0];

// Calculate matrix elements //

          for (i=1;i<flen;i+ +)
          {
                    h[i]=p[i+1]-p[i];
                    d[i]=(q[i+1]-q[i])/h[i];
                    a[i]=h[i-1];
                    b[i]=2(h[i-1]+h[i]);
                    c[i]=h[i];
          }
          for(i=1;i<flen;i+ +)
          {
                    v[i]=6(d[i]-d[i-1]);
          }
// Solve matrix system by Gaussian elimination //

          for(i=2;i<flen;i+ +)
          {
                    f=a[i-1]/b[i-1];
                    b[i] -= fc[i-1];
                    v[i] -= fv[i-1];
          }
```

```
        m[flen-1]=v[flen-1]/b[flen-1];

        for(i=flen-2;i>0;i--)
        {
                m[i]=(v[i]-c[i]m[i+1])/b[i];
        }

// End contraints for natural spline //
        m[0]=0;
        m[flen]=0;
// Calculate spline coefficients //
        for(i=0;i<flen;i++)
        {
                s[i][0]=q[i];
                s[i][1]=d[i]-h[i](2m[i]+m[i+1])/6;
                s[i][2]=m[i]/2;
                s[i][3]=(m[i+1]-m[i])/(6h[i]);
        }
                return(s[flen][4]);
}

void Open_datafile(char road[],long x[],long y[],int rec)
{
        char txt[]=".txt";
        clrscr();

        cout<<"The program assumes that the input files\n";
        cout<<"are of the form <roadname>.TXT\n ";
        cout<<"\n\n"<<"ENTER THE ROAD NAME (e.g. A0257) : ";
        cin>>road;

        strcat(road,txt);
        ifstream fp(road);
        if(fp){
                char inbuf[81];
                cout.setf(ios::fixed,ios::floatfield);
                while(fp.getline(inbuf,81))
                {
                        istrstream ins(inbuf,strlen(inbuf));
                        ins>>x[rec]>>y[rec];
                        (rec)++ ;
        }}
        else
        {
                cout<<"File not found ";
                getch();
                exit(0);
        }
        (rec)--;
        fp.close();
        return;
}

char change_name(char name[],char ext[])
```

```
{
        int i;
        for(i=0;i<12;i++)
        {
                if(name[i]=='.')
                        { name[i]= '\0';}
        }
        strcat(name,ext);
        return(name[12]);
}

// B_SCR.CPP
// Turbo C++
// K. Lupton. 1993
// CALCULATION OF EXISTING ROAD ALIGNMENT USING B-SPLINES

#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <fstream.h>
#include <strstrea.h>

#define N 500
#define nr_steps 3
ofstream fpout;

void Open_datafile(char road[],long x[],long y[],int rec);
char Change_name(char name[],char ext[]);
float B_coeff(float s[][4],long b[]);
void Plot_curve(float sx[][4],float sy[][4],int flen,
                char file[]);

main()
{
        int rec=0;
        long x[N+1],y[N+1];
        float sx[N+1][4],sy[N+1][4];
        char road[12];
        char scr[]="b.scr";

        x[0]=y[0]=sx[0][0]=sy[0][0]=0;
        Open_datafile(road,x,y,&rec);
        B_coeff( sx,x);
        B_coeff( sy,y);
        Change_name(road,scr);
        Plot_curve(sx,sy,rec,road);
        return 0;
}

float B_coeff(float s[][4],long b[])
{
        int i;
        for(i=1;i<=N-2;i++)
```

```
        {
                s[i][0] = -b[i-1] + 3b[i] - 3b[i+1] + b[i+2];
                s[i][1] =  3b[i-1] - 6b[i] + 3b[i+1];
                s[i][2] = -3b[i-1] + 3b[i+1];
                s[i][3] =  b[i-1] + 4b[i] + b[i+1];
        }
        return (s[N][4]);
}

void Plot_curve(float sx[][4],float sy[][4],int flen,char file[])
{
        float t,t1,t2,delt,xpos,ypos;
        int i,k;
        fpout.open(file,ios::out);
        fpout<<"PLINE ";
        for(i=1;i<=flen-2;i++)
        {
                t1=0;t2=1;
                delt=(t2-t1)/nr_steps;
                for(k=1;k<=nr_steps;k++)
                {
                        t=t1+kdelt;
                        xpos=(((sx[i][0]t + sx[i][1])t +sx[i][2])t
                                + sx[i][3])/6;
                        ypos=(((sy[i][0]t + sy[i][1])t +sy[i][2])t
                                + sy[i][3])/6;
                        fpout<<xpos<<","<<ypos<<'\n';
                }
        }
        fpout.close();
        return;
}


// REGSCR.CPP
// Turbo C++
// K. Lupton. December 1993
// CALCULATION OF EXISTING ROAD ALIGNMENT USING
// SMOOTHING SPLINES PARAMETERISED BY CURVE LENGTH.

#include <conio.h>
#include <iostream.h>
#include <stdlib.h>
#include <math.h>
#include <fstream.h>
#include <strstrea.h>
#include <string.h>

const int N= 480;
ofstream fpout;

void  Open_datafile(char road[],long x[],long y[],int rec);
char  Change_name(char name[],char ext[]);
void  Curve_len(long x[],long y[],long ch[],int n);
```

```
float DTD_calc(float V[][7],long c[],int wt,int n);
float B_calc(float B[],float V[][7],long c[],int n);
float Gauss_M(float M[],float B[],float V[][7],int n);
float y_calc(long c[],float M[],float V[][7],int wt,int n);
float S_calc(long c[],float s[][4],float V[][7],float M[],
                int n);
main()
{
        int i,k,j,rec,w,wt;
        long x[N+1],y[N+1],ch[N+1],cpos;
        float V[N+1][7],B[N+1],M[N+1],sx[N][4],sy[N][4];
        float xpos,ypos;
        char road[12];
        char scr[]=".scr";
        V[0][0]=B[0]=M[0]=sx[0][0]=sy[0][0]=ch[0]=0;

        Open_datafile(road,x,y,&rec);
        cout< <'\n'< <"ENTER WEIGHT : ";
        cin> >wt;

                Curve_len(x,y,ch,rec);
                DTD_calc(V,ch,wt,rec);
                B_calc(B,V,x,rec);
                Gauss_M(M,B,V,rec);
                y_calc(x,M,V,wt,rec);
                S_calc(x,sx,V,M,rec);
                DTD_calc(V,ch,wt,rec);
                B_calc(B,V,y,rec);
                Gauss_M(M,B,V,rec);
                y_calc(y,M,V,wt,rec);
                S_calc(y,sy,V,M,rec);
                Change_name(road,scr);
                fpout.open(road,ios::out);

        k=cpos=0;
        fpout< <"PLINE ";
        do {
                for(i=1;i<rec+1;i+ +)
                {
                        if(cpos< =ch[i])
                                { k=i-1;break;}
                }
                        w = cpos-ch[k];
                        xpos=((sx[k][3]w + sx[k][2])w + sx[k][1])w
                                        + sx[k][0];
                        ypos=((sy[k][3]w + sy[k][2])w + sy[k][1])w
                                        + sy[k][0];
                        fpout< <xpos< <","< <ypos< <'\n';
                        cpos += 10;
        } while(cpos< =ch[rec]) ;

        fpout< <x[rec]< <","< <y[rec];
        fpout.close();
        return 0;
```

```
}
void Curve_len(long x[],long y[],long ch[],int n)
{
        int i;
        for(i=1;i<n+1;i++)
        {
                ch[i] = ch[i-1]+hypot(x[i]-x[i-1],y[i]-y[i-1]);
        }
        return;
}
float DTD_calc(float V[][7],long c[],int wt,int n)
{
        int i;
        V[0][3] = c[1]-c[0];

        for(i=1;i<n;i++){
                V[i][3] =  c[i+1]-c[i];
                V[i][0] =  1/V[i-1][3];
                V[i][1] = -1/V[i][3] - 1/V[i-1][3];
                V[i][2] =  1/V[i][3];
                }
        V[n][0]=0;

        for(i=1;i<n;i++){
                V[i][4] =V[i][0]V[i][0]+V[i][1]V[i][1]
                                +V[i][2]V[i][2];
        }
        for(i=2;i<n;i++){
                V[i-1][5] = V[i-1][1]V[i][0]+V[i-1][2]V[i][1];
        }
        for(i=3;i<n;i++){
                V[i-2][6] = V[i-2][2]V[i][0];
        }

        for(i=1;i<n;i++){
                V[i][0] = wtV[i][4] + (V[i-1][3]+V[i][3])/3;
                V[i][1] = wtV[i][5] + V[i][3]/6;
                V[i][2] = wtV[i][6];
                }
        return(V[n+1][7]);
}
float B_calc(float B[],float V[][7],long c[],int n)
{
        int i;
        float prev,diff;
        prev = (c[1]-c[0])/V[0][3];

        for(i=1;i<n;i++){
                diff = (c[i+1]-c[i])/V[i][3];
                B[i] = diff-prev;
                prev = diff;
        }
        return (B[n+1]);
}
```

```
float Gauss_M(float M[],float B[],float V[][7],int n)
{
        float ratio;
        int i;
        for(i=1;i<n-1;i++) {
                ratio = V[i][1]/V[i][0];
                V[i+1][0] -= ratioV[i][1];
                V[i+1][1] -= ratioV[i][2];
                V[i][1] = ratio;
                ratio = V[i][2]/V[i][0];
                V[i+2][0] -= ratioV[i][2];
                V[i][2] = ratio;
        }
// Forward Substitution
        M[0]= M[n]= V[0][2]= 0;
        M[1]=B[1];
        for(i=1;i<n-1;i++) {
                M[i+1] = B[i+1]-V[i][1]M[i]-V[i-1][2]M[i-1];
        }
// Back Substitution
        M[n-1] = M[n-1]/V[n-1][0];
        for(i=n-2;i> =1;i--){
                M[i] = M[i]/V[i][0] - M[i+1]V[i][1] - M[i+2]V[i][2];
        }
        return (M[n+1]);
}

float y_calc(long c[],float M[],float V[][7],int wt,int n)
{
        int i;
        float prev,qu[N+1];
        qu[0]=0;

        for(i=1;i<n+1;i++) {
                qu[i] = (M[i]-M[i-1])/V[i-1][3];
                qu[i-1] = qu[i]-prev;
                prev = qu[i];
        }
        qu[n] = -qu[n];
        for(i=0;i<n+1;i++)
        {
                c[i] -= wtqu[i];
        }
        return(c[n+1]);
}

float S_calc(long c[],float s[][4],float V[][7],float M[]
                ,int n)
{
        int i,j;
        for(i=0;i<n;i++)
        {
                s[i][0] = c[i];
                s[i][1]=((c[i+1]-c[i])/V[i][3])
```

```
                        -V[i][3](2M[i]+M[i+1])/6;
                s[i][2] = M[i]/2;
                s[i][3] = (M[i+1]-M[i])/(6V[i][3]);
        }
        return(s[n][4]);
}

// GSPLINE.CPP
// Turbo C++
// K. Lupton. October 1993
// CALCULATION OF BEZIER POLYGON FOR A SHAPE PRESERVING SPLINE

#include <conio.h>
#include <iomanip.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <fstream.h>
#include <strstrea.h>

#define N 500

ofstream fpout;
char change_name(char name[],char ext[]);
void open_scr(char road[12]);
void h_calc(float c[],float h[],int rec);
void P_calc(float hx[],float hy[],float P[],int rec);
void k_calc(float P[],float k[],int rec) ;
void T1_calc(float T[],float h[],float k[],int rec);
void T2_calc(float T[],float c[],float P[],int rec);

main()
{
        int i,rec=0;
        float x[N],y[N];
        float hx[N],hy[N],P[N],Tx[N],Ty[N],k[N];
        float Vx[N],Vy[N],Wx[N],Wy[N],Jx[N],Jy[N],a,b,c,d;
        char road[12];
        char txt[]=".txt";
        char scr[]="g.txt";

        x[0]=y[0]=hx[0]=hy[0]=P[0]=Tx[0]=Ty[0]=k[0];
        Vx[0]=Vy[0]=Wx[0]=Wy[0]=Jx[0]=Jy[0]=0;

        clrscr();
        open_scr(road);
        strcat(road,txt);
        ifstream fp(road);
        if(fp){
                char inbuf[81];
                cout.setf(ios::fixed,ios::floatfield);
                while(fp.getline(inbuf,81))
                {
                        istrstream ins(inbuf,strlen(inbuf));
```

```
                        ins> >x[rec]> >y[rec];
                        rec+ + ;
        }}
        else
        {
                cout< <"File not found ";
                exit(0);
        }
        rec--;

        change_name(road,scr);
        h_calc( x,hx,rec);
        h_calc( y,hy,rec);
        P_calc(hx,hy,P,rec);
        k_calc(P,k,rec);
        T1_calc(Tx,hx,k,rec);
        T1_calc(Ty,hy,k,rec);
        T2_calc(Tx,x,P,rec);
        T2_calc(Ty,y,P,rec);
        Tx[0]=(3hx[0]/hy[0]-Tx[1])/2;
        Ty[0]=(3hx[0]/hy[0]-Ty[1])/2;
        Tx[rec]=(3hx[rec-1]/hy[rec-1]-Tx[rec-1])/2;
        Tx[rec]=(3hx[rec-1]/hy[rec-1]-Tx[rec-1])/2;
        fpout.open(road,ios::out);

        for (i=0;i<rec ;i+ +)
        {
                a=(hx[i]Ty[i+1]-hy[i]Tx[i+1]);
                b=(hx[i]Ty[i]-hy[i]Tx[i]);

                if(P[i]P[i+1]= =0)
                        {Vx[i]=Vy[i]=Wx[i]=Wy[i]=99;   }

if(P[i]P[i+1]>0)
{
Vx[i]=x[i]+Tx[i]a/((Tx[i]Ty[i+1]-Ty[i]Tx[i+1])+a);
Vy[i]=y[i]+Ty[i]a/((Tx[i]Ty[i+1]-Ty[i]Tx[i+1])+a);
Wx[i]=x[i+1]-Tx[i+1]b/((Tx[i+1]Ty[i]-Ty[i+1]Tx[i])+b);
Wy[i]=y[i+1]-Ty[i+1]b/((Tx[i+1]Ty[i]-Ty[i+1]Tx[i])+b);
}
        if(P[i]P[i+1]<0)
                {
                c=  fabs(Tx[i]Tx[i]+Ty[i]Ty[i]);
                d=  fabs(Tx[i+1]Tx[i+1]+Ty[i+1]Ty[i+1]);
                Vx[i]=x[i]  +  Tx[i]fabs(b)/c;
                Vy[i]=y[i]  +  Ty[i]fabs(b)/c;
                Wx[i]=x[i+1]-Tx[i+1]fabs(a)/d;
                Wy[i]=y[i+1]-Ty[i+1]fabs(a)/d;
                }
                Jx[i]=(Vx[i]+Wx[i])/2;
                Jy[i]=(Vy[i]+Wy[i])/2;
                if(Jx[i]!=99)
                {
                fpout< <x[i]< <" "< <y[i]< <'\n';
```

```
                fpout< <long(Jx[i])< <"  "< <long(Jy[i])< <'\n';
                }
        }
        fp.close();
        fpout.close();
        return 0;
}
void h_calc(float c[],float h[],int fn)
{
        int i;
        for(i=0;i<fn;i+ +)
        {       h[i]=  c[i+1]-c[i];  }
        return;
}
void P_calc(float hx[],float hy[],float P[],int fn)
{
        int i;
        for(i=1;i<fn;i+ +)
        {               P[i]  =  hx[i-1]hy[i]-hx[i]hy[i-1];
        }
        return;
}
void k_calc(float P[],float k[],int fn)
 {
        int i;
        k[1]=fabs(P[2])/(fabs(P[2])+fabs(P[1]));
        for (i=2;i<fn-1;i+ +)
        {
                k[i]=fabs(P[i+1])/(fabs(P[i+1])+fabs(P[i-1]));
        }
        k[fn-1]=fabs(P[fn-1])/(fabs(P[fn-1])+fabs(P[fn-2]));
        return;
}
void T1_calc(float T[],float h[],float k[],int fn)
{
        int i;
        for (i=1;i<fn;i+ +)
        {               T[i]=k[i]h[i-1]  +  (1-k[i])(h[i]);
        }
        return;
}
void T2_calc(float T[],float c[],float P[],int fn)
{
        int i;
        for (i=1;i<fn;i+ +)
                {
                        if(P[i+1]= =0)
                        {       T[i]=T[i+1]=T[i+2]=c[i+2]-c[i]; }
                }
        return;
 }
```