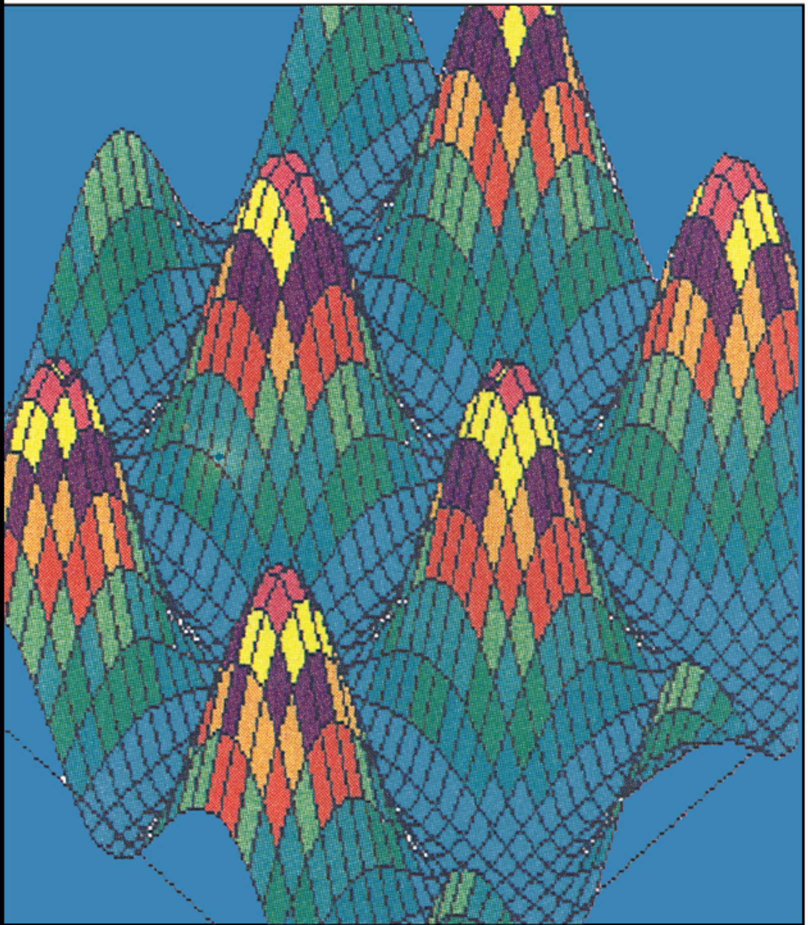

PRESENTATION GRAPHICS

FOR ENGINEERING, SCIENCE AND BUSINESS

P. H. MILNE

E & FN SPON
An Imprint of Chapman & Hall

Also available as a printed book
see title verso for ISBN details

Presentation Graphics for Engineering, Science and Business

Presentation Graphics for Engineering, Science and Business

P.H.MILNE
Department of Civil Engineering
University of Strathclyde, U.K.

E & FN SPON
An Imprint of Chapman & Hall
London • New York • Tokyo • Melbourne • Madras

Published by E & FN Spon, an imprint of Chapman & Hall, 2–6 Boundary Row, London SE1 8HN

This edition published in the Taylor & Francis e-Library, 2005.

“To purchase your own copy of this or any of Taylor & Francis or Routledge’s collection of thousands of eBooks please go to www.eBookstore.tandf.co.uk.”

Chapman & Hall, 2–6 Boundary Row, London SE1 8HN, UK

Van Nostrand Reinhold Inc, 115 5th Avenue, New York NY10003, USA

Chapman & Hall Japan, Thomson Publishing Japan, Hirakawacho Nemoto Building, 7F, 1–7–11 Hirakawa-cho, Chiyoda-ku, Tokyo 102, Japan

Chapman & Hall Australia, Thomas Nelson Australia, 102 Dodds Street, South Melbourne, Victoria 3205, Australia

Chapman & Hall India, R.Seshadri, 32 Second Main Road, CIT East, Madras 600 035, India

First edition 1992

© 1992 P.H.Milne

ISBN 0-203-36214-4 Master e-book ISBN

ISBN 0-203-37472-X (Adobe eReader Format)
ISBN 0 419 15830 8 (HB) 0 419 15840 5 (PB) 0 442 31480 9 (USA)

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the UK Copyright Designs and Patents Act, 1988, this publication may not be reproduced, stored, or transmitted, in any form or by any means, without the prior permission in writing of the publishers, or in the case of reprographic reproduction only in accordance with the terms of the licences issued by the Copyright Licensing Agency in the UK, or in accordance with the terms of licences issued by the appropriate Reproduction Rights Organization outside the UK. Enquiries concerning reproduction outside the terms stated here should be sent to the publishers at the London address printed on this page.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication data available

Contents

Preface

Industry, commerce, finance, government, education, and all scientific fields of technology are depending more on communicating knowledge and ideas through some form of audio or visual presentation. Several colleges and universities (like the University of Strathclyde) now include the subject of Communications in their lecture topics, covering a range of audio and visual presentation techniques. The chapters in this book cover examples for the visual presentation of graphics for engineers and scientists.

The use of a microcomputer (like the IBM PC) to analyse and display graphs and charts, rather than tedious columns of data, has revolutionized modern business reports.

Some twenty-seven different types of graphics presentations are included in the six chapters of this book (Table 6.1). Listings are given for twenty-nine programs, including two menu programs, three data entry programs, fourteen screen display programs and nine plotter conversion programs.

No prior mathematical knowledge of statistics is required to use the programs, and thus the text is suitable, not only for practising engineers and scientists, but also for college or university students who wish to enhance their reports. This book assumes the reader does have access to a microcomputer and is familiar with a BASIC programming language.

Each of the BASIC computer programs in this book can either be run independently by the first time user, or, the programs from any one chapter can be linked together to create a suite of programs. The latter option is only recommended to the experienced computer user with plenty of computer memory to spare. The programs could also be incorporated into a Computer Assisted Learning (CAL) package for students studying statistics.

To simplify the typing of programs, considerable use is made of utility routines, which are described at the end of chapter 1, and the listings given in Appendix A.

Also included in the Appendices are the computer listings used to create the figures used for illustration, together with a list of the global variables used. Example computer printouts are presented, together with computer graphic screen dumps to both dot-matrix and ink-jet printers and also plotters.

As some readers may wish to convert these programs to run on alter native microcomputers to the IBM PC, a conversion table of graphics commands is included. All printout routines follow a standard format with no ESCape codes, so they should run on any ASCII or IBM PC compatible printer. Users of my previous book *Computer Graphics for Surveying* will find they can now import their contour data files for display in full colour, either as shaded contour maps or as surface models.

I am extremely grateful to Professor A.McGown, the current chairman of the Department of Civil Engineering at the University of Strathclyde for his encouragement in publishing the results of my research work. I would also like to thank Mrs Sheena Nelson of the Mechanical, Civil and Chemical Engineering Drawing Office at the University of Strathclyde for assistance in the preparation of the hand-drawn figures in the book.

I am grateful to both Mr Michael Dunn and Mr Nick Clarke for helpful contact and advice in editing the book, especially in connection with the graphics presentations and computer listings.

Finally I would like to thank my family for their patience, support and understanding during the preparation of the book. I would also like to thank my two sons, Robert (studying Information Engineering at University) and Gordon, for assistance with the computer programming. This book could not have been completed without the wholehearted support of my wife, Helen, whom I thank most sincerely for her helpful suggestions and perseverance in typing the complete manuscript.

Department of Civil Engineering,
University of Strathclyde,
John Anderson Building,
Glasgow, G4 0NG,
Scotland.

P.H.Milne,
April 1991.

PROGRAM DISC AVAILABLE

This book contains many lengthy program listings once all the routines have been added. To save typing them into your computer, the programs are available on $3\frac{1}{2}''$ and $5\frac{1}{4}''$ discs for the IBM PC, PS/2 and compatibles. For details of prices, etc., and availability of other microcomputer software/hardware formats, readers should contact the author.

TRADEMARKS

Please note the trademarks used in this book are the property of the companies with which they are associated, e.g. HPGL is a trademark of Hewlett-Packard Corp., IBM is a registered trademark of International Business Machines Corp., Macintosh is a trademark of Apple Computers.

While every effort has been made to ensure the accuracy of the information given in this publication, the Author and Publishers will not be liable if the state of scientific and technical knowledge at the time of sale was not such that they might have been expected to discover any defect in the software. Neither the Author nor the publishers in any way accept liability for loss of any kind resulting from the use or misuse made by any person of such information.

Glossary of abbreviations and technical terms used with microcomputers and presentation graphics

Abscissa The horizontal or X-axis of a statistical grid.
Align To arrange letters, words etc. on the same vertical axis.
Annotation An explanatory note forming part of an illustration.
Area chart A graph chart that represents a quantity by the area under a line. Each series in an area chart is represented by a layer, the base of which is the previous series.
Arithmetic mean The sum of the values recorded in a series of observations divided by the number of observations.
Array An arrangement of elements (numbers, characters, etc.) in rows and columns.
ASCII American Standard Code for Information Interchange. A standard that assigns a specific code to each of 128 digits, letters, and control characters.
Average A measure of the most 'typical' value in a series of observations. There are three ways of expressing averages: arithmetic mean, median or mode.
Axis A fixed line adopted for reference. Graphs or charts are usually organized on axes which are at right angles to each other.
Bar chart A form of pictorial presentation where bars are used to provide comparison between items.
Base line The imaginary line on which the data stand, e.g. the zero line in a bar chart or histogram.
BASIC Beginner's All-purpose Symbolic Instruction Code. A high level computer language, with most commands in recognizable English.
Baud rate The rate of data transmission, often designed so that one baud equals one binary bit of data.
BASICA IBM BASIC for the IBM PC.
Best-fit This is a straight line, drawn using linear regression, through a set of data, where the same number of points or co-ordinates lie on each side of the line.
Border A single or double line that surrounds a complete chart.
CAL Computer assisted learning.
Color/Graphics Adapter (CGA) The lowest graphics resolution available on an IBM PC or compatible, providing a choice of four colours. Requires a Color/Graphics Adapter card.
Character The individual letter, numeral or punctuation mark. Note: when counting characters and calculating the space they will occupy, it is essential to count inter-word spaces as one character.
Chart A drawing containing text or a graph.
Chart name The name used to store a chart on, and retrieve it from, a disc. A chart name is equivalent to a DOS filename and must conform to DOS rules for filenames, i.e. no more than 8 characters with a three character extension.
Coefficient of variation The standard deviation divided by the mean.
Column chart A chart containing one or two vertical bars in which each column is divided into slices by horizontal lines.
Column width The width of one column of data in a column chart or type as in books and journals.
Compatible In hardware, the ability to work with or act in an identical manner to another piece of equipment. In software, the ability to interchange files or data without the need to re-enter them from the keyboard.

Contour A line joining points of equal value (normally height) to plot isograms.
Coordinate A precise reference, which locates a point, line or plane, in 2 or 3 dimensional space.
Correlation Whether or not there is any association between two variables.
Cross hatching The criss-cross patterns made by tiling to simulate textures.
Cumulative graph (or chart) A graph chart in which each point represents the sum of all values up to that point.
Current chart The chart in the computer's memory.
Cursor The square on the screen that indicates where the next character typed will appear.
Curved line chart A variation of the trend line chart in which a curved line passes through the graphing area.
Cut slice A slice in a pie chart that is 'exploded' (moved slightly away from the rest of the pie). In a pie chart, a cut slice is used for emphasis.
DGM see **Digital ground model**.
Default A setting or value that the software will use unless you change it.
Dependent variable A variable which is altered by changes in the independent variable. Dependent variables should always be placed on the vertical axis of a graph.
Digital ground model (DGM) A digital representation of relief (ground surface).
Disc A device used for the storage of information on a permanent or semi-permanent basis. See also **Disc drive**.
Disc drive A device which contains a reading and writing head for loading data onto a disc, or reading data from a disc. Hard discs of much greater storage capacity are usually housed in sealed units, whereas flexible discs or floppies are easily swapped.
Dispersion Measures of deviation or spread around a central point.
Dump Transfer amounts of data straight to a peripheral, like a printer or disc.
Enhanced/Graphics Adapter (EGA) This card provides a greater graphics resolution with a choice of sixteen colours.
Ellipse An oval: if the pie chart looks elliptical on the graphics screen, change the value stored in **ASPECT**.
Export To save on disc a graphics chart in order to use it with another program.
File An organized collection of related records. The records on a file may be related by a specific purpose, format, or data source, and the records may or may not be arranged in sequence. A file may be made up of records, fields, words, bytes, characters or bits.
Fill To display or draw a character in solid colour rather than as an outline, or to draw a shape with a centre that is a solid colour or pattern.
Fish-net mesh A visualization technique for representing a surface in 3-D by plotting lines through each grid node.
Floppy disc see **Disc**.
Frequency If a set of data is divided into categories, the number of items in each category is known as the frequency distribution.
Graph A 'grid' on which curves are plotted to illustrate the relationship between two variables.
Grid lines The lines that mark the unit of measurement horizontally, and sometimes vertically, across a graph or chart.
GW-BASIC A version of Microsoft BASIC for use on an IBM PC, PS/2 or compatible.
Hardcopy A printed paper copy of a program or its graphic results produced by a printer or plotter connected to the microcomputer.
Hardware Generic term for all manufactured computer equipment, i.e., the physical parts as contrasted with the programs (software).
Histogram A stepped column chart, without gaps between the columns, in which the area of each column represents a frequency distribution. The horizontal scale represents types of occurrences or ranges of size, and the vertical scale represents frequency of occurrence. Charts drawn as histograms are useful for analysing cumulative distributions of data.
Horizontal chart Typically, a bar chart in which the bars run horizontally and the uses of the *X* and *Y* axes are reversed.
Horizontal format see **Landscape**.

HPGL Hewlett-Packard Graphics Language, a set of two-letter mnemonic commands by means of which a computer controls a plotter or other output device.

Import To retrieve from disc either data or a chart, created outside the current software program.

Independent variable Any variable whose values are not affected by changes in other variables. Time is an example. Independent variables are normally placed on the horizontal axis of a graph.

Interface A device for linking one component with another, such as a printer, plotter or digitizing tablet and a microcomputer, to permit transfer of data.

Label A word, phrase, or other text that identifies a slice in a pie/ column chart or a series in a bar/line chart.

Landscape The orientation of a chart that is displayed or printed down the length of the page, i.e. with the *X*-axis along the longest dimension, instead of across the page, as is normally done (in portrait orientation).

Least-squares method Produces a line drawn through the data which minimizes all positive and negative deviations of the data from this line.

Legend The patterns, markers, or colours (with accompanying labels) that identify the series in an area chart, bar chart or line graph.

Linear trends Methods of arriving at a linear or straight line trend.

Logarithmic A scale type for the *X* and/or *Y* axes that uses base 10 logarithms for the numeric divisions along the axes. The distance between divisions decreases as you go up the scale.

Log-log A graph chart in which both the *X* and *Y* axes are scaled logarithmically.

Loop A sequence of instructions repeated until the loop is terminated.

Macro A macro can be defined as the capability to combine many actions, e.g. keystrokes, into one simple command.

Marker A symbol used to show a data value (or point) in a line or point chart.

Mean deviation The arithmetic average of all the differences between the observations and their mean.

Median The value of the middle item of a distribution, or series of observations, which is arranged in ascending order, e.g. 1 1 1 2 3 3 4 5 6 7 9, median 3.

Memory Any device used to store data or instructions for the computer. Memory devices are compared in terms of storage capacity, access time and cost.

Menu A list of options presented to the operator during execution of a program.

Monitor A cathode ray tube (CRT) display screen for text and/or graphics, often called a VDU or visual display unit.

Monochrome In one colour only.

MS-DOS Microsoft disk operating system.

Multiple chart A single chart that displays up to five sets of data on a single page, screen, or slide.

Ordinate The vertical or Y-axis of a graph or chart. Can also be any vertical line which bisects the abscissa.

Orientation The direction of the longest dimension of an object or illustration.

PAINT see Table 1.1

Parallel communications The standard character and ASCII code transmission method where bits are sent on eight lines at a time in parallel, normally used for printer communications.

Palette The overall selection of colours or shades available.

Pattern A design used to fill an area such as slices in a pie/column chart, bars in a bar chart, or layers in an area chart.

Perspective A three-dimensional view.

Pie chart A graph chart in which a circle is divided into slices by straight lines. The circle represents the total, or whole amount, and each slice represents a part of the whole.

Pixel The smallest addressable picture element or point on a VDU, generally given as a number of horizontal and vertical points, e.g. 640×200 for **SCREEN 2**.

Plotter Hardcopy device with a resolution much superior to that of a graphics screen with straight diagonal lines.

Point chart A graph chart in which each pair of *X* and *Y* data values is shown as a point, unconnected by a line.

Portrait The orientation of a chart that is displayed or printed across the page in an upright mode, i.e. with the *Y*-axis vertical.

Program A list of computer instructions connected in a logical format directing the computer to perform specific operations.

QuickBASIC Another version of BASIC from Microsoft for IBM PC, PS/2 and compatibles with advanced features; user defined data types, recursion, sub-programs, flexible array dimensioning, merging of files, compiling and a good editor.

Range The difference between the lowest and highest values observed.

Regression Attempts to show the relationship between two variables by providing a mean line which best indicates the trend of the points or coordinates on a graph.

Resolution The degree of detail that can be produced on a screen, printer, or plotter.

RS-232C Serial communications interface for plotters etc.

Scale The range of values covered by the *X* or *Y* axis of a graph chart.

Scatter chart see **Point chart**.

Screen dump An exact replica of the graphics screen on a dot-matrix or colour printer.

Screenshow A slide show displayed on a computer screen.

Serial communications The standard character and ASCII code transmission method where bits are sent, one at a time, in sequence, normally used for plotters and digitizing tablets.

Series A set of data, which when displayed, represents a pie or column, a single line (in a line chart), a set of bars (in a bar chart), or a layer (in an area chart). A single chart can contain data for up to five different series.

Slide A chart produced by a screen capture program and incorporated in a slide show.

Slide show A list of charts, templates, and other files used to create batch output, practice cards, or a **Screenshow**.

Software Generic term for computer programs and digitized information which is used to issue instructions to the computer hardware and peripherals for specific applications.

Stack To display a chart so that each series uses the previous series as its base.

Standard deviation The square root of the sum of the square of the deviations of the individual values from the mean of the distribution, divided by the number of items in the distribution.

Standard error of the mean The square root of the arithmetic mean of the squares of the differences between the observations and their mean, plus one.

Statistics Is concerned with scientific methods for collecting, organizing, summarizing, presenting and analysing data, as well as drawing valid conclusions and making reasonable decisions on the basis of this analysis.

String This is a sequence of characters (letters or numbers, or a combination of both) that begins and ends with double quotation marks.

Surface model A 3-D representation or visualization of the surface by plotting grid data as an isometric projection.

Tabulation The systematic arrangement of data into columns.

Template A pre-defined chart, with default settings, for the location of axes, data and text.

Tiling A method of designing patterns on the screen where the colour attributes of each pixel can be varied within a given boundary, rather than plotted in a single solid colour. To use tiling, the **PAINT** attribute must be a string expression rather than a single colour.

Transfer format The format used to transfer data between computer systems.

Trend line chart A line chart in which a straight line, determined by linear regression, passes through the graphing area.

Value In a pie/column chart, the quantity or percentage contributed by a slice to the whole.

Variable Data subject to measured change.

Variance The average of the square of the deviations.

Vertical format, see **Portrait**.

Video Graphics Array (VGA) This is a higher graphics resolution than EGA, but cannot be used to display graphics written in BASIC. It will, however, emulate EGA.

Visual display unit (VDU) see **Monitor**.

Window The currently displayed portion of the screen used for the display of graphics.

Wireframe see **Fish-net mesh**.

***X* Axis** Normally, the horizontal axis of a graph chart, which shows the way the data is classified. (In a bar chart, the *X* axis is the vertical axis).

***Y* axis** Normally, the vertical axis of a graph chart, which shows the quantity or amount. (In a bar chart, the *Y* axis is the horizontal axis).

Zigzag line chart A line chart in which straight lines connect the data in each series.

100% chart A variation of the stacked bar chart that shows the percentage contribution of each series to the whole.

1
Introduction to presentation graphics

1.1
INTRODUCTION

This chapter is an introduction to the use of presentation graphics, with specific reference to the graphics capabilities of the IBM PC, PS/2 and their compatibles. It explains the advantages of interactive computer graphics displays for the presentation of different types of graphs and charts for use by engineers, scientists and statisticians from all disciplines. In addition to screen presentations it is also important to look at printer and plotter output.

Also provided is an overview of the workings of the computer graphics hardware and software of the IBM PC, PS/2 or their compatibles, using an interactive high level language like BASICA, GW-BASIC or QuickBASIC, to create and control the graphics output.

1.2
PRESENTATION GRAPHICS

The main aim of presentation graphics is to communicate statistical information, and the type of presentation will therefore depend on the requirements and interests of the people receiving the information. Just as a picture can be worth a thousand words, a graph or chart can be worth a thousand numbers. Those numbers or figures need to be arranged and presented in some specific format before the information contained in the data can be interpreted.

The various features to consider are:

(i) clear presentation of the subject matter;
(ii) clarification of the most important points in the data;
(iii) consideration of the purpose of the presentation;
(iv) consideration of the amount of detail and accuracy required;
(v) the use of the most appropriate method for presentation.

Several methods of data presentation can be used with a choice from:

(i) written descriptions,
(ii) tabulation,
(iii) line graphs,
(iv) 3-D graphs,
(v) pictorial presentation, e.g.—bar charts, pictograms, pie charts, strata charts,

(vi) frequency distributions e.g. histograms.

The principal advantage of presentation graphics over the tabulation of data or report writing is that a graphical presentation can often discover previously undetected correlations or errors in the data. Indeed the use of graphics can be more precise and more revealing than conventional statistical computations.

The type of chart chosen to communicate the information will depend on the nature of the data. In general there are five major classifications of data which can be conveyed by graphs or charts, as shown in the gallery of examples in Fig. 1.1.

1.2.1
Trend or time series

This is the most commonly used type of chart. It displays a single data series over a period of time to show trends. It can also be used to compare two or more series of data over the same period. Trend or time series charts can take several forms:

Trend/time series chart	*Chapter*
Line graph	2
Bar chart (horizontal)	3
Column chart (vertical)	3
Multiple bar/column chart	3

1.2.2
Comparisons of different items

This type of chart emphasizes direct comparisons of unrelated items at one point in time, or over a period of time. Where comparisons are being made, the chart can take several forms:

Comparison chart	*Chapter*
Multiple bar chart (horizontal)	3
Multiple column chart (vertical)	3
Area (strata) chart	2

1.2.3
Comparisons of components

These are charts which define the pieces of data which make up a complete unit. Pie charts (sometimes called circle or sector charts) are the most commonly used. However, stacked (or segmented) bar/column charts are very valuable for showing the absolute size of an element as well as for the comparison of totals. Component charts can take several forms:

Component chart	*Chapter*
Pie chart	4
Stacked bar/column chart	3
Area chart	2

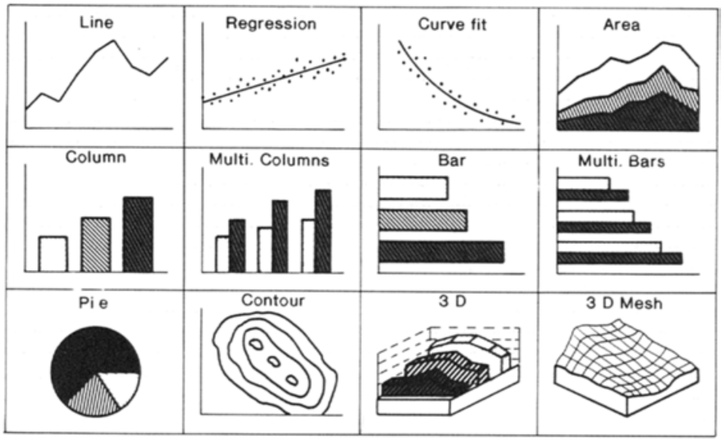

Fig. 1.1 Examples of the different types of graphs and charts, which can be displayed by the programs in this book.

Component chart	*Chapter*
100% stacked bar/column chart	3

1.2.4
Representations of correlation

These graphs indicate the relationship between *dependent* and *independent* variables, measured as correlation or regression coefficients. The resulting regression or trend line can then be overlaid on the data, e.g. scatter plot, for analysis. Correlation graphs can take several forms:

Correlation graph	*Chapter*
Line graph	2
Linear regression	2
Exponential curve	2
Power curve	2
Cubic spline	2

1.2.5
Other chart forms

Also available in many computer graphics packages are Gantt or PERT charts used for project management (too specialized for this book), and Word charts (again not dealt with here, as these are often available in word processing and desktop publishing packages). Various 2-D and 3-D charts are, however, included to allow a better visualization and insight into the data. The following types of charts available are:

Other chart forms	*Chapter*
2-D line contours	4
2-D shaded contours	4
3-D column charts (vertical)	5
3-D area charts	5
3-D profiles	5
3-D open mesh models	5
3-D shaded mesh models	5

Once data has been saved on disc, (see Section 1.10), the user can experiment with the type of chart to obtain the optimum presentation graphics display for that data set. For example, if the data for a chart are for one item over a consecutive period of time, and details for individual years are to be stressed, a column chart would be the first choice. However, assuming that the given space for the chart is higher than it is wide, a more readable layout would be a bar chart format.

1.3
COMPUTER LANGUAGE

To be able to use computer graphic commands, it is essential to use an advanced language such as BASIC or C. The programs in this book are written in advanced BASIC and will therefore run under BASICA, GW-BASIC or QuickBASIC on an IBM PC, PS/2 or compatible. Although the language BASIC itself can be used to plot points and draw lines, it is very crude and time consuming. The three BASIC languages mentioned, from Microsoft Inc., include several graphics commands as listed in Table 1.1 which simplify the programmer's task. There are considerable advantages in using QuickBASIC to run the programs, for example:

(i) it has a very good editor for cutting and pasting;
(ii) it provides a compiler to speed up the computation times and reduce screen drawing delays;
(iii) it will handle larger data sets than BASICA or GW-BASIC; and
(iv) the experienced programmer will be able to compile the utility routines (see Section 1.9) into 'include' files similar to that found in the C language.

Table 1.1 Graphics commands

CIRCLE	Draws circles, ellipses and arcs
CLS	Clears the screen
COLOR	Sets the colour of plotted points and the background colour
DRAW	Draws using a graphics control language
LINE	Draws line between given end points
PAINT	Fills an area of the screen with colour
PSET	Sets points on the screen
SCREEN	Selects the graphics mode for the screen
VIEW	Sets a viewport on the screen
WINDOW	Defines the logical size of the screen window

1.4
HARDWARE AND SOFTWARE

To be able to display on screen any of the computer graphics programs discussed in this book, the user will require a graphics adapter, either CGA, EGA or VGA. To get the best performance from the presentation graphics, a colour monitor with either an EGA or VGA graphics adapter card are recommended. It is not essential to have a colour monitor as the user is presented with a choice of colour or monochrome displays at the installation time. If the user has a CGA adapter, the displays will be in monochrome. However, if the user has only a monochrome adapter card for text, then a plotter will be required to obtain a hardcopy of the graphics output, Fig. 1.2.

All the programs presented in the book allow the user to obtain a text printout of the data and results. It is assumed that a parallel printer interface is available, using the address "LPT1:". If the user has a graphics adapter and a dot-matrix printer, then it is possible to dump the graphics screen output to the printer for record purposes. In this case the program "graphics" has to be loaded from PC-DOS or MS-DOS before running BASICA, GW-BASIC or QuickBASIC.

Whereas the IBM PC graphics display commands will run on any IBM PC or compatible under BASICA, GW-BASIC, or QuickBASIC, there is unfortunately no standard graphics plotter language. However, many manufacturers have adopted the Hewlett-Packard Graphics Language (HPGL) for their plotters, and HPGL is used in the programs in this book.

Although the programs can be run from a single floppy disc microcomputer, a dual floppy or hard disc machine with one floppy disc is recommended. With a dual floppy machine, the master BASIC or compiled programs should be run from Drive A: and the data stored on Drive B:. If a hard disc machine is used, better performance will be achieved with the master BASIC or compiled programs on Drive C: in a graphics

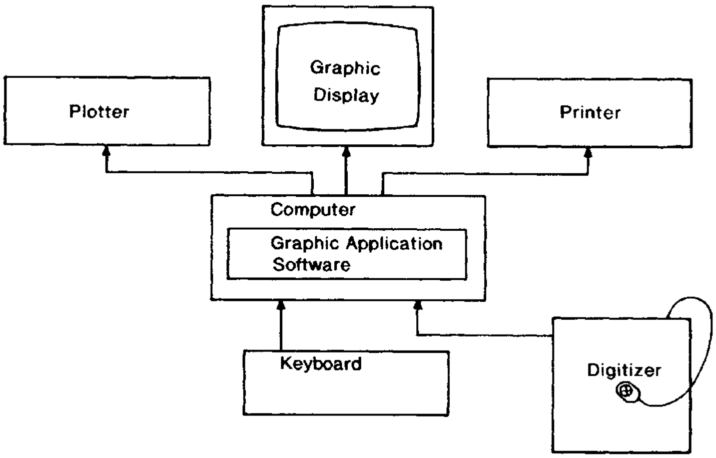

Fig. 1.2 Typical computer graphics system. The graphic application software is the heart of the system accepting and processing input from the keyboard (or digitizer) with output generated on a graphic display (colour/monochrome), printer and plotter.

directory, (e.g. **C:\PGRAPHS**). The user then has a choice of storing the data, either on a floppy disc, Drive A:, or if space is available on the hard disc, in a data directory (e.g. **C:\PGDATA**).

Table 1.2 Screen graphics modes

Screen	*Command*	*Horizontal resolution*	*Vertical resolution*	*Colours*
CGA	**SCREEN 1**	320	200	4
CGA	**SCREEN 2**	640	200	2
EGA	**SCREEN 9**	640	350	16

1.5
PERFORMANCE CHARACTERISTICS

Microcomputer users are often confused by the terminology used to describe computers, as 8-bit, 16-bit, 32-bit or 64-bit. This refers to the number of data lines connected to the systems microprocessor, i.e., either 8, 16, 32 or 64. One would therefore expect 32-bit and 64-bit computers to run much faster than 8-bit or 16-bit machines, and in general this is true.

The late 1980s saw an upsurge in the number of microcomputer chips released, first the 86, followed by the 286, 386 and 486, each with the capability of running much faster than their predecessor. Associated with this has been the ‘tweaking’ of the chips to run at faster speeds from the original 4.77 MHz through 16 MHz up to 25 MHz and 33 MHz.

All the programs described in these pages will run on any IBM PC, PS/2 or compatible microcomputer from the original IBM PC at 4.77 MHz (86 chip) up to the latest 33 MHz (386/486 chip). The time taken to display a specific presentation graphics chart will therefore depend on the computational speed of the microcomputer and its associated video board. If the 2-D and 3-D colour shading programs described in Chapters 4 and 5 are being run regularly, then the user should choose a microcomputer with a faster MIPS (million instructions per second) rating. Note that it is the MIPS rating rather than the MHz rating which is more important; for example a 33 MHz PC with a 386 chip will operate at approximately 6 MIPS whereas a 25 MHz PC with a 486 chip will operate at approximately 12 MIPS.

As screen graphics are very much faster than plotter graphics, it is suggested that users when given a choice, check the output on the screen before using the plotter. Some plotters incorporate buffers and can be run at a baud rate of 9600 to speed up the plotting, but some require a slower rate, e.g. 1200 baud, and thus large A3 or A1 contour drawings can take a considerable length of time. Some of the faster microcomputers, running at over 8 MHz are too fast for some plotters and give a “Device timeout error”. If this happens, a delay routine has to be incorporated into the program, as described in Section 1.9.23.

1.6
SCREEN GRAPHICS

As detailed in Table 1.2 it is possible to plot on the screen using different graphics modes. The user in this book has to choose between colour (EGA/VGA) or monochrome (CGA). The medium-resolution CGA colour mode **(SCREEN 1)** only allows a resolution of 320 points across the screen and 200 points down the screen whilst the high-resolution CGA monochrome **(SCREEN 2)** has 640 points across and 200 points down the screen. With the EGA colour mode **(SCREEN 9),** the user can access 640 points across and 350 points down the screen. As it is not possible to use VGA graphics modes from BASIC programs, VGA screens use EGA emulation.

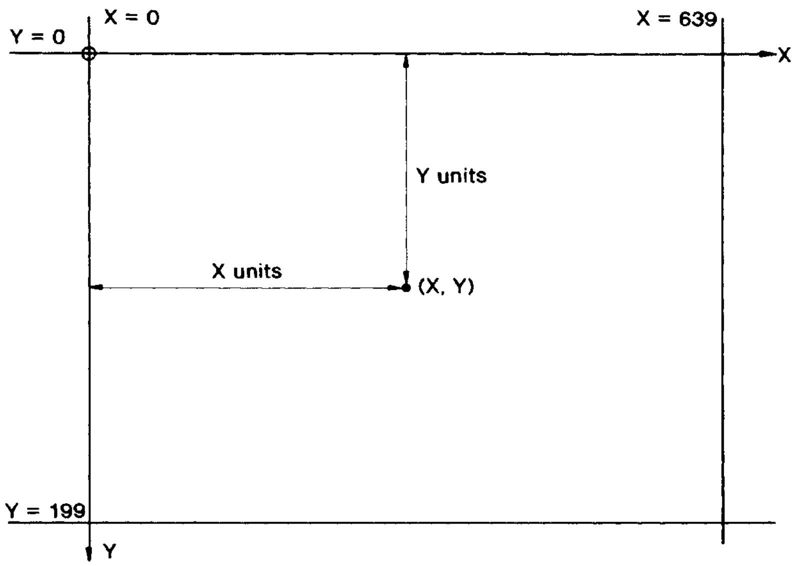

Fig. 1.3 Default graphics screen coordinates with zero (0, 0) in top left-hand corner of screen.

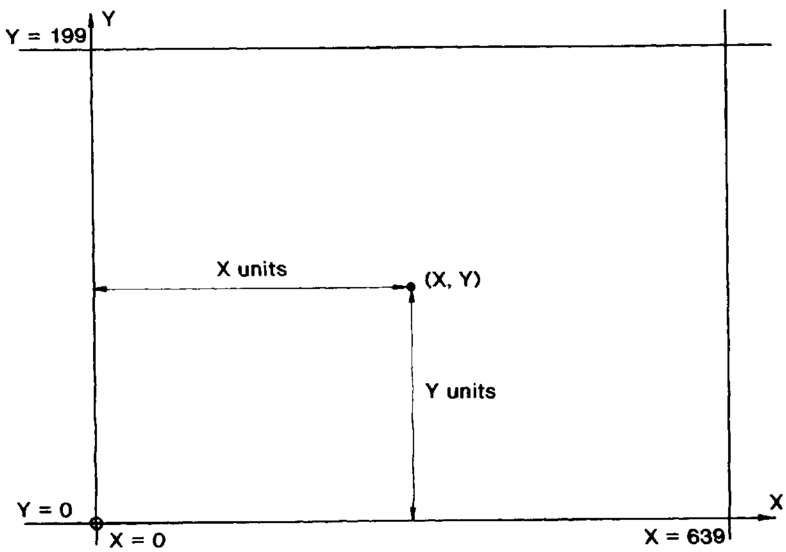

Fig. 1.4 Normal plotting screen coordinates with zero (0, 0) in lower left-hand corner of screen to be compatible with digitizers and plotters.

Note that reference is made to measuring across and down the screen. This is because the zero **(0,0)** is in the top left-hand corner of the screen. The maximum *Y*-value is either 199 points with CGA or 349 with EGA, but the *X*-value will remain constant at 639 irrespective of whether **SCREEN 2** or **SCREEN 9** graphics are being used. Although it is possible to use the CGA medium resolution mode with 4 basic colours from Table 1.2, the graphics resolution is very poor and hence the use of the high-resolution (monochrome) mode. To simplify the graphics plotting in the programs,

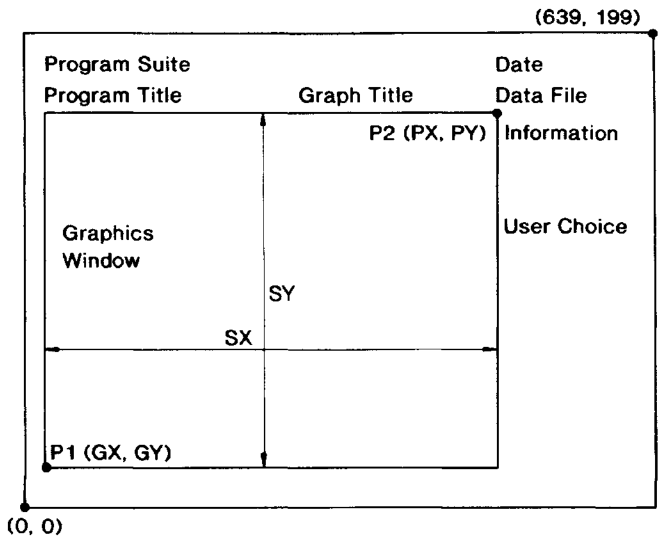

Fig. 1.5 Combined graphics window and text screen with text at top and right-hand side of graphics display.

the user is given a choice of selecting at the outset, either CGA or EGA graphics, as discussed in the next section. The graphics screen in Fig. 1.3 would therefore be set up using the command:

WINDOW SCREEN (0,0) – (649,199)

However, in general plotting it is usual to have the zero in the bottom left-hand corner, and this is the way all plotters are normally set up. Accordingly in the programs the zero of the graphics screen is taken to be in the bottom left-hand corner of the screen (see Fig. 1.4) by using the command:

WINDOW (0,0) – (649,199)

As the above commands define the whole area of the screen, it is often desirable to shrink the graphics screen to allow text to be displayed either at the top and right-hand side of the screen, Fig. 1.5, or at the top and bottom of the screen, Fig. 1.6. Examples of both types of screen displays are to be found in this book. In the first case it is desirable to leave two text lines clear at the top of the screen. To enable these programs to be transferred to other microcomputers, (Table 1.3), with different resolutions, the view screen coordinates are referred to using variables, i.e. left-hand corner P_1 (X-min, Y-min) as **(GX0,GY0)** and right-hand corner as P_2(X-max, Y-max) as **(PX,PY)** where the size of the view screen is given by **SX** and **SY**. The variables are then defined at the start of the program, e.g.

GX0=70 : GY0=50 : SX=500 : SY=120

This plotting area then covers three quarters of the horizontal width of the screen with graphics allowing text characters at the right-hand side with two lines at the top for text, e.g., title headings, date, time, etc., Fig. 1.5. It is suggested that title headings, etc., are stored in string variables for easy recall. One advantage of the IBM PC and its compatibles is that it has a built-in calendar/clock, and if **DATE$** is used as a variable, it returns the date in the form MM-DD-YYYY, i.e. in the US date format. Similarly if **TIME$** is used as a variable, it returns the time from the built-in clock in the form HH:MM:SS, thus allowing files and computer printouts to be date and time stamped for future reference.

If, however, a wider plotting width is required, then the second choice, Fig. 1.6, should be used and a small border left at either edge of the screen, for example:

VIEW SCREEN(20,20) – (620,180)

with two lines for text at the top for the headings, etc., and two lines at the foot for instructions and user choices.

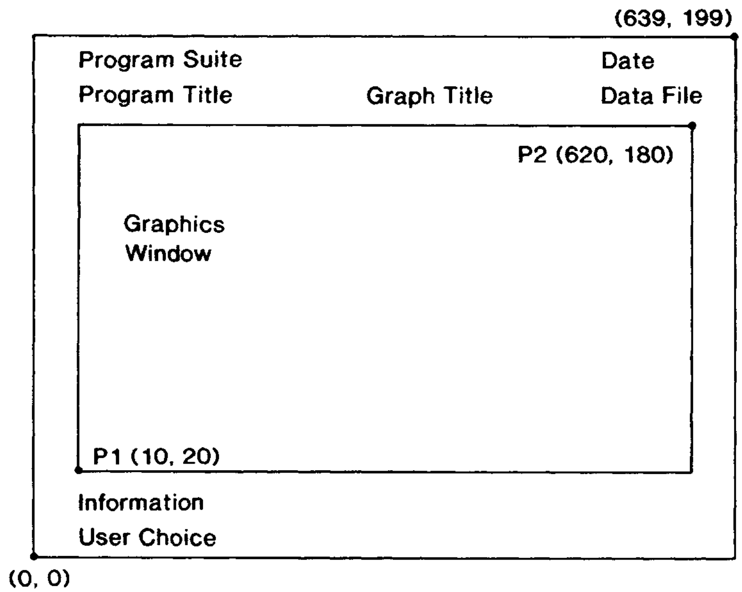

Fig. 1.6 Combined graphics window and text screen with text at top and bottom of graphics display—useful for wider plots and multiple user choices.

Table 1.3 Comparison of computer graphics commands

IBM PC, PS/2	*Apple Macintosh*	*Apple II series*	*BBC B*
CLS	**CLS**	**HOME**	**CLS**
COLOR	**Forecolor** **Backcolor**	**COLOR**	**COLOUR**
LINE	**CALL LINETO**	**HPLOT**	**DRAW, PLOT**
PAINT	**CALL PAINTRECT**	–	**PLOT** [80–87]
PSET	**CALL MOVETO**	**HTAB, VTAB**	**MOVE, PLOT**
SCREEN	–	**HGR, HGR2**	**VDU** [codes]
VIEW	–	–	**VDU** [codes]
WINDOW	**WINDOW**	–	**VDU** [codes]

To delineate the graphics plotting area, the user has a choice of drawing a box **(B)** round the area either using the **LINE** command, such as

LINE (GX0,GY0) – (PX,PY), RGB%, B

where **RGB%** is a variable for the required colour; or alternatively using the **VIEW SCREEN** command, such as

VIEW SCREEN (20,20) – (620,180),, RGB%

The two commas before the selected colour for the box, **RGB%,** indicates that the area is to be left blank; if a number is inserted here, the box will be filled with the designated colour.

If a colour monitor is being used, the choice of background and text colour is most important. These programs have been written to give white text on a blue background in colour or a black background in monochrome to ensure legibility.

If red is used in **SCREEN 9,** its number must be changed when a monochrome monitor is used, as black would not be seen on monochrome, hence the storing of the colour in the variable **RGB%** which allows it to be changed depending on the screen resolution. It should also be pointed out that program lines with the command **COLOR** must be avoided if using a monochrome monitor with **SCREEN 2** as they will generate an "illegal function" error.

All the programs in this book clear the screen using the **CLS** command, and as no editing takes place during the running of the programs, the special function key display which normally appears at the foot of the screen is removed with the command **KEY OFF**.

If the user has a dot-matrix graphics printer, it is possible to obtain a screen dump of the graphics screen display. This requires the MS-DOS routine "graphics" to be loaded prior to loading BASICA, GW-BASIC or QuickBASIC. Then, to obtain a screen dump, the user can press the [Shift] and Print Screen [Prt sc] keys simultaneously. Alternatively, if a graphics printer is not available, the graphics screen can be saved to disc with several general utility programs described in Chapter 6. As most of the graphics screens for engineering use the *landscape* (horizontal) format, rather than the *portrait* (upright) format (see Glossary for definitions), the landscape format is used for all the graphs/charts and their screen dumps from **SCREEN 2** (CGA monochrome) and **SCREEN 9** (EGA colour). The recommendations of the British Standards Institution (DD 52:1977) for graphics presentations have also been followed.

1.7
GRAPHICS INSTALLATION

To simplify the selection of graphics screens from within the programs in this book, a graphics installation program **"PGINSTAL"** is included to allow the user to store on disc the preferred options for colour or monochrome and, where the master program and data files are stored on disc. Once the user has selected the correct configuration, a data file, **'PGSCRDSK.PGD',** is saved to disc for recall by subsequent graphics programs. If the user has only a monochrome text adapter, no graphics displays will be possible, but hardcopies will still be available on a plotter. Details of this program are given in Appendix A.1.

1.8
PLOTTER GRAPHICS

Due to the relatively low resolution of the screen graphics of the IBM PC and its compatibles, diagonal lines often show characteristic stair-stepping effects. To produce good hardcopies of screen graphics, it is often desirable to use a graphics plotter. There are numerous models on the market, some of which use a natural plotting language, whilst others like Hewlett-Packard use HPGL (Hewlett-Packard Graphics Language). This is an industry standard and other manufacturers often offer the user the choice of HPGL emulation, e.g., Benson, Hitachi and Roland. For those unfamiliar with HPGL, it is a set of two-letter mnemonic commands by which a computer controls a plotter or other output device. The syntax of HPGL commands has a standard format, with the mnemonic followed by parameters necessary to specify it. The mnemonic and parameters are separated by delimiters (,) and the whole command is completed with a terminator character (;).

As mentioned previously, the zero **(0,0)** of all plotters is in the lower left-hand corner. Distances are therefore measured from the lower and left-hand edges of the plotter. The resolution of a plotter is, however, much superior to that of a graphics screen. For example, the horizontal and vertical point resolution of an A4 plotter can be 10800 by 6800 and of an A3 plotter, 15200 by 10800 points respectively. Each plotter unit therefore represents 0.025mm, i.e., there are 40 plotter units per millimetre. When a plotter is turned on, it is normally initialized to specify the parameters of P_1 (*X*-min, *Y*-min) and P_2 (*X*-max, *Y*-max). For example, if using a popular A3 plotter like the Hewlett-Packard HP-7475 A, the initialization command would be:

"IP 0,0,15200,10800;"

giving a total of 15200 plotter units in the *X* direction and 10800 plotter units in the *Y* direction. This plotting area can then be scaled into user units to match the graphics screen plotting coordinates used by P_1 and P_2 in Fig. 1.5, such as

"SC";GX0;",";PX;",";GY0;",";PY;";"

The line plotting routine is however more complex than that discussed previously, such as

LINE (GX0,GY0) – (PX,PY), RGB%,B

To plot the same rectangle on an HP-7475A plotter requires the following commands:

"PU;PA";GX0;",";GY0;";"
"PD;PA";PX;",";GY0;";"
"PA";PX;",";PY;";"
"PA";GX0;",";PY;";"
"PA";GX0;",";GY0;";PU;"

where **"PU"** is for Pen Up, **"PA"** for Pen Absolute and **"PD"** for Pen Down. The programs in this book give the user the option of three different interface connections, either a parallel interface **"LPT1:" or "LPT2:"** or an RS-232C interface with address **"COM1:",** i.e., the primary asynchronous communications interface. It is up to the user to enter the correct plotter communication setting at the appropriate line number. It is suggested that to save repetitive keyboard work the plotter communication setting is stored at Line 50000, and the plotter initialization commands are stored in routines at Lines 51000, etc., as described in Section 1.9.18.

1.9
UTILITY ROUTINES

To simplify the task of writing the computer programs described in Chapters 2, 3, 4 and 5, some twenty basic utility procedures or routines were written for incorporation into the programs as required. Thus it is only necessary to type the routine once and save it on disc under an appropriate heading for merging into the presentation graphics programs for recall, using a **GOSUB** command.

As discussed earlier in the Preface, many CAL features which make software portable also make it easy to implement, such as:

(i) use of a well-known language, e.g., BASIC;
(ii) minimal use of non-standard features of the language;
(iii) modular design, which keeps coding in one place, and enhances the understanding of the coding itself.

Attention to these details should lead to a package which is easy to use, especially if combined with clear diagnostic error messages when a routine is unable to function.

Another desirable feature is the sensible choice of default values for parameters, such as the positioning of axes on a graph, or the range which the coordinates of a graph may cover. If default values for parameters are set, an unpractised programmer using the software can avoid the problem of having to make an initial choice until more familiar with the software.

1.9.1
Check on graphics screen and disc set up

This routine saved at Line 8000 opens the data file **'PGSCRDSK.PGD'** saved in Section 1.7 and recalls the screen resolution CGA or EGA, and whether in colour or monochrome. The location of the master programs and the data files, discussed in Section 1.4 are then recalled and the computer screen graphics set up accordingly. The computer listing for this routine is given in Appendix A.2.

1.9.2
Recall of data file

8200

This routine saved at Line 8200 opens the data file **"IDFILE.PGD"** saved when data are entered as discussed in Section 1.10. The name of the data file **'DFILEN$'** and its extension **'.EXT$'** is recalled and the user has a choice of loading this data file or specifying another data file. The selected data file is then opened and the number of rows and columns of data with their respective data values are stored in memory. This computer listing is given in Appendix A.3, and consists of two segments of code starting at Lines 8200 and 8800.

1.9.3
Writing of text file to disc

8400

Either at the time of entry of data, Section 1.10, or at the subsequent recall of a data file it is possible to store on disc the title of the specific Presentation Graphics chart together with the *X*-axis and *Y*-axis labels. Where several data items are stored on disc, each set of data can also be labelled accordingly. This routine writes a data file with the extension **'.TXT'** and is stored at Line 8400. This computer listing is given in Appendix A.4.

1.9.4
Recall of text file from disc

8500

This routine saved at Line 8500 opens the **'.TXT'** file saved in Section 1.9.3 and stores the graph title, axes and data labels in string variables for plotting on the graph or chart. This computer listing is given in Appendix A.5.

1.9.5
Data printout routine

6000

Once graph or chart data have been entered, as described in Section 1.10, this routine allows the user to obtain a hardcopy printout of the data for reference, including any associated text file as described in Section 1.9.3. This routine is saved at Line 6000 and the computer listing is given in Appendix A.6.

1.9.6
Data analysis printout routine

6500

Where a single set of data has been analysed and a curve fitted through the data, e.g., linear regression, exponential or power curve fit, this routine at Line 6500 will give a printout of the results. Both the graph factors *A* and *B* together with the *X, Y* data points, the theoretical value of *Y* and the residual or variance are printed for analysis. The computer listing for this routine is given in Appendix A.7.

1.9.7
Data check for maximum and minimum values

7000

This routine, stored at Line 7000, runs a check through the data sets to find the maximum and minimum values of *X* and *Y* so that the graph axes can be set correctly. The computer listing for this routine is given in Appendix A.8.

1.9.8
Graph axes intervals

7100

The handling of graph axes and scales is always a compromise between maximizing the plotting area, but at the same time presenting a uniform appearance. If the scales are adjusted automatically by the computer, using the maximum and minimum values from Section 1.9.7, the graph may give distorted information or provide a misleading appearance, especially as the vertical axis should always start at zero. After the user has chosen the range of data to be plotted, this routine will check the data set and ensure, no matter the size of the data range, that only 10 subdivisions are plotted on both the *X* and *Y* axes. To avoid confusion, in this book the terms 'abscissa' and 'ordinate' will not be used. Instead, all graphs will be plotted with the *independent* variable scaled on the *X*-axis (horizontal) and the *dependent* variable scaled on the *Y*-axis (vertical). The only variation to this convention is where a horizontal bar chart is drawn. This routine, which can handle sub-divisions from 0.01 up to 500, is stored at Line 7100 and the computer listing is given in Appendix A.9.

1.9.9
Set up graphics screen template

7300

Once all the data have been entered and adjusted accordingly, this routine will clear the graphics screen and set up the necessary axes and labels depending on the type of graph being plotted, e.g., pie chart, bar chart or line graph. This routine is saved at Line 7300, and the computer listing is given in Appendix A.10.

1.9.10
Display graph text labels

7500

Reference was made in Sections 1.9.3 and 1.9.4 to the ability to annotate the graph with a title and labels for axes and data. This routine places the graph title centrally at the top of the screen, with the *X*-axis label and data below the graph. The *Y*-axis label can be printed horizontally or printed vertically to the left of the *Y*-axis. This routine is stored at Line 7500 and the computer listing is given in Appendix A.11.

1.9.11
Display graph axes grid

7600

It is often desired to plot data against a background grid for reference, especially where data is to be interpolated from a line graph. This routine, saved at Line 7600, gives the user the option to plot a horizontal grid of dotted lines separately, or in conjunction with a vertical grid. The computer listing for this routine is given in Appendix A.12.

1.9.12
Set up colour shaded panel

7800

Where colour shading is used to indicate height on a contour or surface model chart, this routine stored at Line 7800 will display each of the colours and associated levels on the right-hand side of the screen. The computer listing for this routine is given in Appendix A.13.

1.9.13
Set up monochrome shading

8600

Where bar charts or area charts are being presented on monochrome screens, this routine is provided to set up different types of hatching using the process known as *tiling*. Each tile for a pattern is composed of a rectangular grid of pixels, and five different patterns are supplied. This routine is saved at Line 8600 and the computer listing is given in Appendix A.14.

1.9.14
Screen header

9000

This routine is used to annotate each computer display, either graphics or text, with the title of the software suite —"Presentation Graphics", together with the date, the type of graph and the data file being displayed. This routine is saved at Line 9000 and the computer listing is given in Appendix A.15.

1.9.15
Program end

9500

This routine is run at the end of each program and allows the user to run the program again or return to the master menu. This routine is saved at Line 9500 and the computer listing is given in Appendix A.16.

1.9.16
Error routines

10000

Numerous errors can cause a computer program to crash, e.g., the user has forgotten to insert a floppy disc into the data drive, or has forgotten to load a data file before trying to run a display program. These routines saved at Line

10000 should cater for most of the straight-forward errors. Where a user has modified the program and introduced a new error, the program will display a note of the type of error and its line number to identify its location. The computer listing for the error routines is given in Appendix A.17.

1.9.17
Data symbol display

40000

To identify different sets of data, five different symbols are supplied: a square, a triangle, a diamond, a cross sign and a plus sign. This routine, saved at Line 40000, is accessed by the line graph, area chart and bar chart displays to identify the colours or monochrome hatching used. The computer listing is given in Appendix A.18.

1.9.18
Plotter set up

50000

This routine is included to allow users to set up an HPGL or compatible plotter for hardcopy output. A choice of interfaces (**LPT1:, LPT2:** and **COM1:**) are given as well as two paper sizes, A4 and A3. This routine is saved at Line 50000 and the computer listing is given in Appendix A.19.

1.9.19
Plotter title

50500

This routine is used to annotate each plotted graph with the title of the software suite, together with the date, the type of graph and the data file being displayed. This routine is saved at Line 50500 and the computer listing is given in Appendix A.20.

1.9.20
Plotter line specification

51000

This routine is used to choose different pen colours and line types for each data set, e.g. dashed, dotted, chain-dotted, etc. using the **LT** command. This routine is saved at Line 51000 and the computer listing is given in Appendix A.21.

1.9.21
Plotter shading specification

52000

This routine is used to choose different types of hatching for bar charts and pie-charts, e.g. cross-hatched, diagonal, etc. using the **FT** command. This routine is saved at Line 52000 and the computer listing is given in Appendix A.22.

1.9.22
Graphics numbers

55000

One of the problems with the IBM PC and its compatibles is that in labelling axes, etc., the **PRINT** statement uses a text location rather than a graphics location on the screen. This means that if the graph axes labels used the **PRINT** function, the numbers would not line up with the ticks on the axes and would tend to overwrite other data. To overcome the problem, this routine included at Line 55000, draws each number at its specified location on the graphics screen. The width **(W)** and the height **(H)** of the numbers can be varied by changing the variables accordingly. This computer listing is given in Appendix A.23.

1.9.23
Delay routine for plotters

60000

As discussed in Section 1.8, some microcomputers send data so quickly that it can be lost in transmission to the plotter. To prevent this happening a 'Delay' routine should be included in the program. This routine is saved at Line 60000 and the computer listing given in Appendix A.24. The choice of a high program line number is to keep it clear of other program lines. The value of **TSEC** is set before calling the subroutine, and will depend on the particular combination of microcomputer and plotter being used.

1.10
DATA PREPARATION

Three data preparation programs are included in this section to cater for data entered either at the keyboard or recalled from disc.

1.10.1
Keyboard data entry

This small program allows the user to enter data from the keyboard. The essential features required are a data file name **(DFILEN$),** the number of data sets **(ND)** and the number of data points **(NROW)**. The data are then stored in a matrix **Z(I,J)** where **I** represents the number of rows **(NROW)** and **J** represents the number of columns **(NCOL)**. Note **NCOL** equals **ND** plus one to allow for the *X*-axis data to be stored in the first column. The data are entered, one item at a time, at the foot of the screen and a data table displayed on screen. After ten rows of data have been entered, the screen is cleared to allow another page of data to be viewed. On completion, the data is stored in a disc file, **'DFILEN$',** with the extension **'.DAT'**. An identity file **'IDFILE.PGD'** is also saved to disc for recall by the graphics display programs described in subsequent chapters. The user can also enter a **'.TXT'** file at the same time, as described earlier in Section 1.9.3, to save the graph title, etc. Having entered the data and text, the user can obtain a printout of the data and text as described earlier in Section 1.9.5. On completion, the user can select the type of graph or chart for display from the master menu. The computer listing of this program, **"KEYBDATA",** is given in Appendix A.25 and utilizes several of the routines described in Section 1.9.

1.10.2
Disc data recall

In general, initial data input to the programs is via the keyboard, with data files being saved to disc for access by other programs. However, if ASCII data files have been saved from a spreadsheet program, like Lotus 1-2-3 and Microsoft Excel, these can be read by the program **"DISCDATA"** and saved using the **'.DAT'** extension as described in Section 1.10.1. The user can specify the extension for the ASCII file, e.g. **'.PRN'** or **'.ASC',** so files saved by any spreadsheet or MS-DOS editor can also be read. The format to be followed in preparing such a file is a header containing the number of rows and number of columns, followed by one line for each row of data, i.e. *X*-axis value plus the *Y*-values for each data set—each numeric item being separated by at least 2 blank or null spaces. As before, a text **'.TXT'** file can also be saved for display purposes. The computer listing for this disc data entry program **"DISCDATA"** is given in Appendix A.26 and utilizes several of the routines described in Section 1.9. This program also saves an identity file, as described in Section 1.10.1 so it can be used to reload data previously saved to disc by the keyboard entry routines. Data and text file printouts are also available.

In addition to spreadsheet data files, it is also possible to load 2-D and 3-D data files either generated by the program in Section 1.10.3 or from my previous book, *Computer Graphics for Surveying,* where contour data files were stored with a **'.CON'** extension. These files can be loaded and displayed using the 2-D contour and 3-D surface modelling programs described in this book.

1.10.3
Data generation program

The previous data entry file (Section 1.10.1) was for data where the *Y*-values of a data set were plotted against the *X*-axis. There are often situations where a function z is known in terms of *x* and *y* and it is desired to explore the generated surface of that function either as a 2-D contour plot or as a 3-D surface model. This program, **"KEYBEQUD",** allows the user to define any function in terms of *x* and *y,* e.g:

$$f(z) = \sin(x^2 + y^2) + \sin(y) \tag{1.1}$$

between user limits, i.e. *X*-min, *X*-max and *Y*-min, and *Y*-max. The resulting values of the functions are then stored in a matrix **Z(I,J)** where **I** is the number of rows and **J** the number of columns, normally each 41 to keep the computation time reasonable. The resulting data file is stored with a **'.EQD'** extension, and the relevant text in a **'.TXT'** file as before. The complete computer listing is given in Appendix A.27 where the function to be evaluated is stored in Line 250. The programs described in later chapters thus allow the user to plot the resulting surface on screen or plotter in a similar fashion to that obtainable in complex mathematical programs like 'Mathematica'.

1.11
PLOTTER ROUTINES

To save storage space and to simplify the transfer of screen graphics programs to a plotter, several of the previous screen routines have been converted for plotting, as listed below:

Section	*Details*	*Line No.*	*Appendix*
1.9.9	Graphics template	7300	A.28
1.9.10	Text labels	7500	A.29
1.9.11	Background grid	7600	A.30
1.9.12	Bar chart template	7800	A.31

Section	*Details*	*Line No.*	*Appendix*
1.9.17	Data symbol display	40000	A.32

If each of the screen graphics programs are saved as a **"GRAPH-"** program, e.g. **"GRAPH-B1"** for the program in Appendix B.1, then the plotter conversion program should be saved as **"GPLOT-B1"**.

2
Line graphs and area charts

2.1
INTRODUCTION

2.1.1
Line graphs

Line graphs are useful for describing and comparing trends in numerical data over time. They are most effective for showing a change in one or more sets of scientific data over time, especially if the data changes dramatically from period to period.

To compare the trends of several items, a multiple line chart can be used. Although it is possible to show up to eight sets of data in a single chart, the results can be very difficult to interpret. In general, the number of data series in a chart should not exceed five. However, if any of the lines cross, a chart can be very confusing if more than three or four data sets are included. Rather than have five confusing lines on one chart, it is preferable to split the data between two charts and combine the data sets which show similar trends.

Line graphs should therefore be chosen when:

(i) data covers a long period of time;
(ii) several series of data are compared on the same chart;
(iii) emphasis is on the trend rather than the actual amount;
(iv) trends of frequency distribution are presented;
(v) estimates, forecasts, interpolation or extrapolation are required.

The three types of line graphs commonly used to present trend or time series data are:

zigzag line—uses straight lines to connect the data in each series and is the most common and easiest graph to interpret, and can be used for multiple data sets. It is most important that the data is sorted and saved with an increasing *X*-value.

trend line (or scatter plot)—here the individual data points are not connected by the trend line; instead a best-fit trend line is drawn through the data points.

curve line—this is a variation of the trend line graph where a smoothed cubic spline is passed through each of the data points for any one set of data. As with the zigzag line, data has to be sorted before saving with an increasing *X*-value.

When plotting line graphs, the user has also to consider the best method of presentation, either as

Table 2.1 Types of line graph

Type of graph	*Description*	*Advantages*	*Disadvantages*
Natural scale graph	Illustrates the relationship between one variable and another	Relationships can be seen; widely used and well understood; interpolation and extrapolation are possible	Only certain data can be shown on a graph
Straight line graph	Shows the relationship when there is direct variation between two variables	Possible to interpolate information and use as a ready reckoner	Shows a particular relationship
Semi-log graph	Shows rate of change of data	Curve shows rate of change rather than magnitude	Shows a particular relationship; can be confusing

(Adapted from Hannagan (1986))

(i) a natural scale graph,
(ii) a straight line graph, or
(iii) a semi-log graph.

Only the first two types are discussed in this book as the third type can be confusing, see Table 2.1.

2.1.2 Area charts

Area or strata charts are similar to line graphs, but have different connotations. A line graph measures the change in data from one point to the next, whereas an area chart emphasizes the volume of the data, from the baseline of the chart to the top of the trend line. There are three types of area or strata charts:

(i) *cumulative* area chart,
(ii) *stacked* area chart, and
(iii) *100% stacked* area chart,

as discussed in Section 2.3.

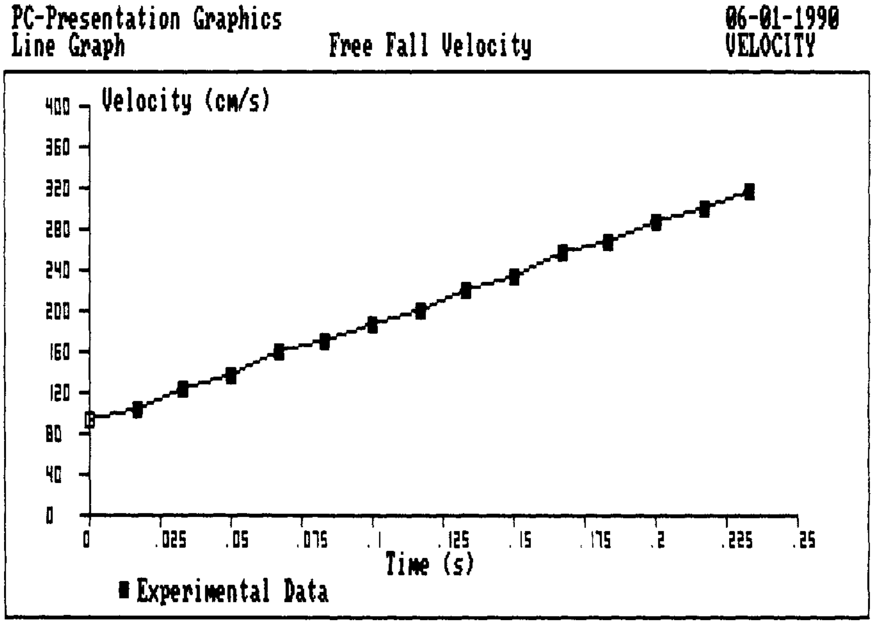

Fig. 2.1 Zigzag line graph—single data set: plot of experimental data to study uniformly accelerated motion of a freely falling weight by comparing velocity against time.

2.2
LINE GRAPHS

2.2.1
Zigzag line graph
Appendix B.1

A zigzag line graph is suitable for a single data set or multiple data sets where measurements have a similar range of values along the *X*-axis. To differentiate between multiple data sets, each set of data is plotted with a different line type (e.g. solid, dashed, chain-dotted, etc.) and a data point marker (square, diamond, triangle, cross or plus). The markers and lines are also plotted in a different colour if using a colour screen.

Many computer programs clutter up the screen with function key labels and command lines, thus spoiling screen dumps. The computer listing of a program to display a zigzag line graph, Fig. 2.1, is presented in Appendix B.1.

The main features of this program are that the last data file prepared is automatically recalled. The user can choose this or select another data file. Once this is loaded, the relevant *X* and *Y* ranges are displayed on a text screen for alteration as required. Data for line graphs should be prepared as outlined in Chapter 1, and saved to disc. The essential features of the data file, with a **'.DAT'** extension will be a header indicat ing the number of data points, i.e. **NROW,** up to a maximum of 100, and the number of columns **NCOL,** up to a maximum of 6, i.e. 5 data sets. If the user wishes to increase the number of data points, and correspondingly reduce the number of data sets, to keep within the memory limitations of the program, the array variable **Z(100,6)** in Line 80 of the program should be altered accordingly. Graph axes should always meet at zero with no breaks in the scales to prevent distortion. Neither should the scale for the axes be compressed so that the slope is artificially distorted. Occasionally, however, where the data fill a narrow band of values remote from a zero, it is permissible to alter one of the scales, provided that the starting value is shown clearly.

As discussed in Chapter 1, routines are provided to set up the graph axes from the chosen maximum and minimum values of X and Y, and also the axes labels. Before display, the user can choose how to present the data, with a choice from

(i) line chart with lines and markers,
(ii) lines only,
(iii) markers only,
(iv) lines and markers with horizontal grid, and
(v) lines and markers with horizontal and vertical grid.

The third choice, markers only, can be used for scatter plots of data where there is no correlation. The latter two choices with grids are very useful for time series charts, where the user wishes to stress exact values or interpolate values from the line graph. As discussed in Chapter 1, routines are provided to plot differently shaped markers and provide horizontal and vertical background grids. The graph is displayed after setting up the graph axes, graph labels and display requirements. On completion, the user can obtain a screen dump by pressing the [Shift] and [Prt sc] keys simultaneously, Fig. 2.2.

The zigzag line provides a pictorial record of the data, but does not provide a statistical analysis of the data or plot a trend line, as discussed in the next three sections.

2.2.2 Trend line graph

Appendix B. 2, 3, 4

Trend line graphs are most frequently used with a single statistical data set to show the relationship between two variables. This book presents the following options:

(i) linear regression (least squares),
(ii) exponential curve fit, and
(iii) power curve fit.

In any discussion of statistical data, there are several terms used to analyse the deviation or spread of data, normally called measures of dispersion, see Table 2.2. The most commonly used terms are range, mean deviation and standard deviation. However, in a statistical analysis, the variance and coefficient of variation are extremely useful for comparing distributions between different data sets.

In comparing statistics, regression and correlation are concerned with whether or not there is any association between two variables.

Linear regression

Appendix B. 2

Regression refers to an attempt to show the relationship between two variables by plotting a mean line which best indicates the trend of the coordinates on a graph. This straight line is based on the equation

$$y = a + bx \qquad (2.1)$$

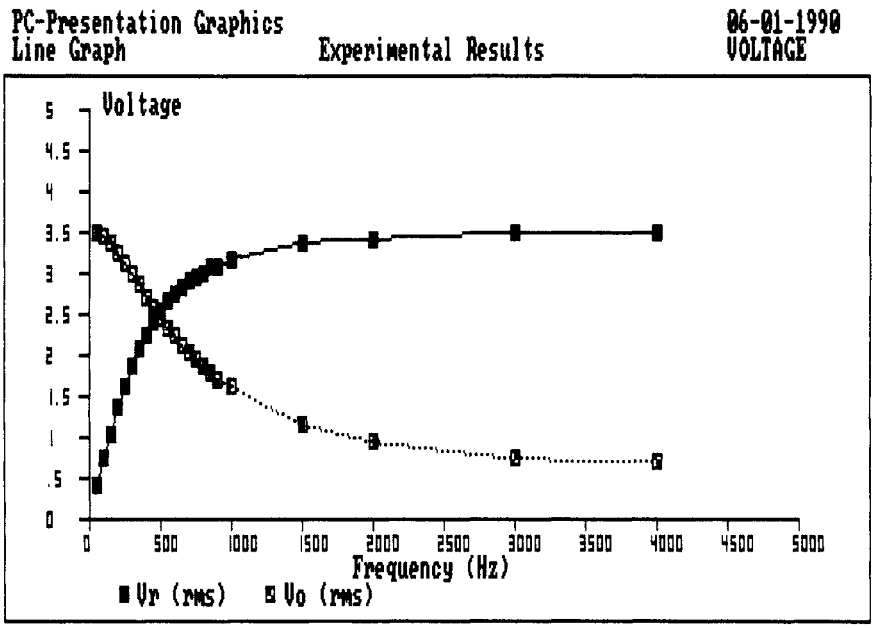

Fig. 2.2 Zigzag line graph—multiple data set: plot of the voltage (V_r) across a resistor compared with the voltage out (V_0) for a range of frequency.

where *x* and *y* are the two variables, *a* represents the intercept on the *Y*-axis and *b* represents the slope of the line, as shown in Fig. 2.3. In the past, a statistician had to plot a line of best fit by eye through a scatter plot of all the data points. Where there is a strong linear correlation between two variables, the best mathematical solution is the least squares method. The aim of this method is to produce a line which minimizes all positive and negative deviations of the data from a straight line drawn through the data. This is achieved by squaring the deviations and therefore a least squares regression line is the line which minimizes the error in the deviation of the variable being predicted. The slope of the line *b* for *n* data points is found first from

$$b = \frac{n\,\Sigma xy - \Sigma x\,\Sigma y}{n\,\Sigma x^2 - (\Sigma x)^2} \tag{2.2}$$

The intercept *a* is then found from

$$a = y_\mathrm{m} - bx_\mathrm{m} \tag{2.3}$$

where x_m and y_m are the *arithmetical means* of the two variables.

$$x_\mathrm{m} = \frac{1}{n}\,\Sigma x \tag{2.4}$$

$$y_\mathrm{m} = \frac{1}{n}\,\Sigma y \tag{2.5}$$

The computer listing for the linear regression program is given in Appendix B.2. The data entry format and display are the same as described for zigzag line graphs. The user has a choice of plotting the data points with or without a background grid as well as the linear regression line, Fig. 2.4. A data output file with a **'.LIN'** extension saves the values of *a* and *b*. In addition to the screen display, the user can also obtain a graphics screen dump and a printout of the

Table 2.2 Measures of dispersion

Measures of dispersion	*Description*	*Advantages*	*Disadvantages*
Range	Highest value minus lowest	Everyday measure, clear, does not depend on number of items	Provides limited information, can be 'distorted' by extreme items
Mean deviation	Arithmetic mean of the absolute difference of each value from the mean	Uses all the values in a distribution	Seldom used except in statistics
Standard deviation	Dispersion of items around the arithmetic mean	Uses all the values in a distribution, used in sampling and in other mathematical areas	Difficult to understand, not an everyday measure
Variance	Square of the standard deviation	Uses all the values in a distribution	Difficult to understand
Coefficient of variation	Standard deviation divided by the mean	A measure of relativity for comparing distributions	Difficult to understand

(Adapted from Hannagan (1986))

statistical analysis, giving the measures of dispersion; arithmetic mean, mean deviation, standard deviation, variance and correlation coefficient, where the standard deviations are given by

$$s_x = \sqrt{\frac{\Sigma x^2 - n x_m{}^2}{n - 1}} \tag{2.6}$$

$$s_y = \sqrt{\frac{\Sigma y^2 - n y_m{}^2}{n - 1}} \tag{2.7}$$

with the variance of

$$x = S_x^2 \tag{2.8}$$

and the variance of

$$y = S_y^2 \tag{2.9}$$

The correlation coefficient is calculated from

$$r_{xy} = \frac{\frac{1}{n-1}\left(\Sigma xy - \frac{1}{n}\,\Sigma x\,\Sigma y\right)}{s_x s_y} \tag{2.10}$$

Correlation is concerned with whether or not there is any association between two variables. It is useful to know:

(i) whether any association exists;

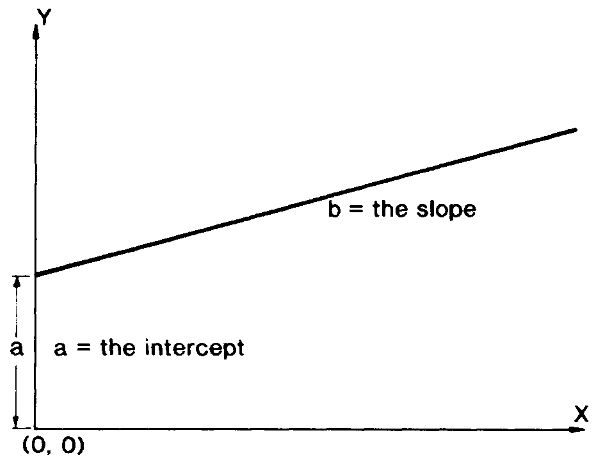

Fig. 2.3 Trend line graph—linear regression: relationship between *a* the intercept and *b* the slope in Eqt. 2.1. The slope of the line shows changes in one variable against the other.

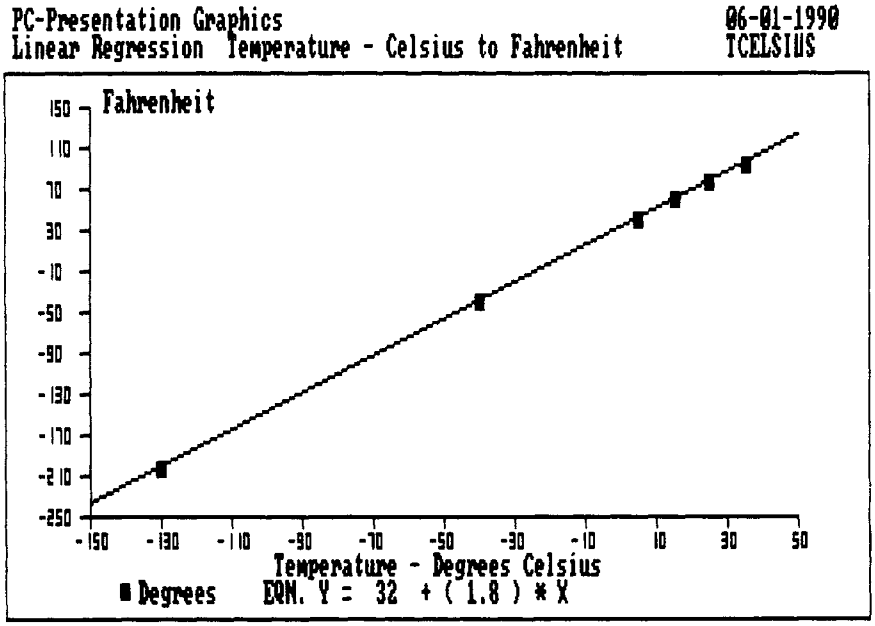

Fig. 2.4 Trend line graph—linear regression: plot of temperature—degrees Celsius against degrees Fahrenheit showing linear regression equation solution on screen.

(ii) the strength of the association, from −1 (strong negative association) to • 0.1 and +0.1 (weak association) to +1 (strong positive association);

(iii) the direction of the relationship; whether it is positive (in the same direction) or negative (in the opposite direction);

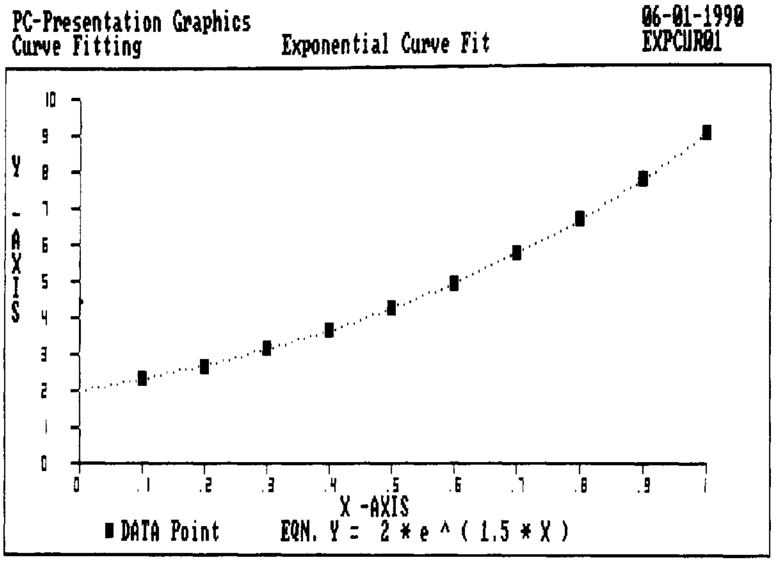

Fig. 2.5 Trend line graph—exponential curve fit: plot of exponential data given by Eqt. 2.11 showing equation solution.

(iv) the proportion of the variability in one variable that can be accounted for by its relationship with the other variable; if r_{xy} (the correlation coefficient) equals +0.9 then r_{xy}^2 equals +0.81, and it can then be said that 81% of the variability in one variable can be accounted for by its linear relationship with the other variable.

The coefficient of determination, which is the square of the correlation coefficient, indicates the quality of fit achieved by the regression. Values of r_{xy}^2 close to 1.00 indicate a better fit than values close to zero. The regression coefficients a and b which define the line generated are also printed.

Exponential curve fit

Appendix B.3

Often data collected do not follow a linear relationship, for example an exponential curve fit where

$$y = ae^{bx} \tag{2.11}$$

Note that a must be greater than zero to obtain a solution. The methods of data entry, display and results are the same as described for linear regression, and the computer listing for this program is given in Appendix B.3. As before, both the data points and the exponential curve can be plotted with and without a background grid, Fig. 2.5. A data output file with a **'.LIN'** extension saves the values of a and b for future use.

Power curve fit

Appendix B.3

Many compression processes can be correlated using a power curve of the form

$$y = ax^b \tag{2.12}$$

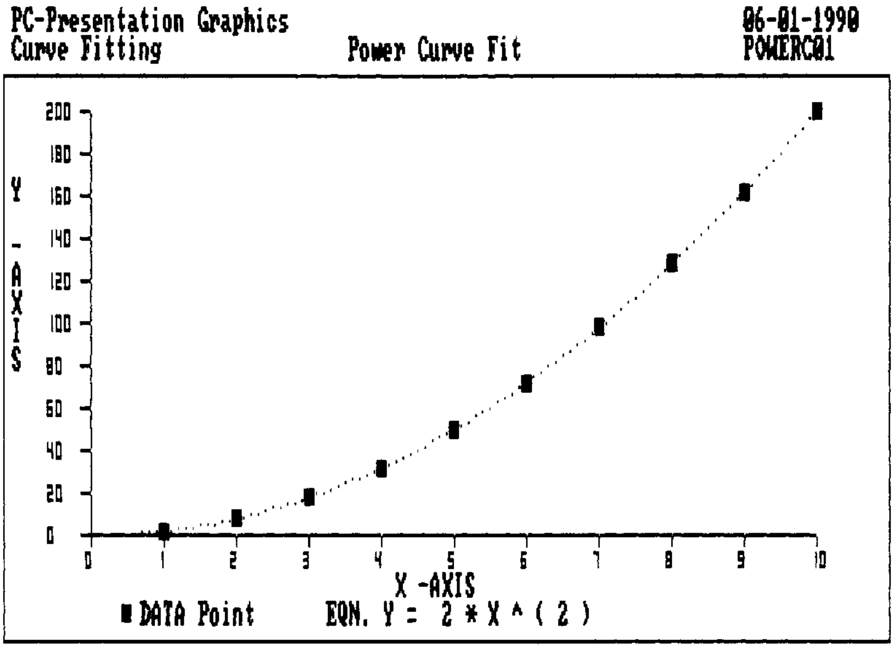

Fig. 2.6 Trend line graph—power curve fit: plot of power curve data given by Eqt. 2.12 showing equation solution on screen.

If the above equation was for pressure-volume data y would be the pressure, x the volume and b the polytrophic constant. Note that a must be greater than zero to obtain a solution. The method of data entry, display and results are the same as described for linear regression, and the computer listing is given in Appendix B.3. As before, both the data points and the power curve can be plotted with or without a background grid, Fig. 2.6. A data output file with a **'.LIN'** extension saves the a and b factors for future use.

2.2.3 Curved line

Appendix B.4

This variation of the trend line graph uses a smoothed cubic spline to pass a curve through each of the data points. The theory of splines is very extensive, covering Bezier curves and *B*-splines and is outside the scope of this book. In general a spline is a collection of polynomials guaranteed to be as smooth as possible. It is convenient to represent splines as parametric functions of a parameter t where

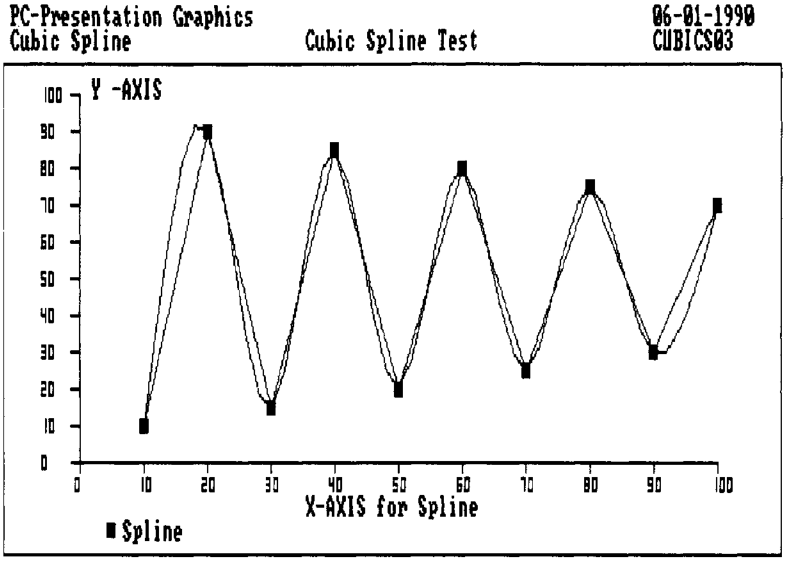

Fig. 2.7 Trend line graph—curved line: plot of smoothed cubic spline passing through each of the data points using Eqt. 2.15.

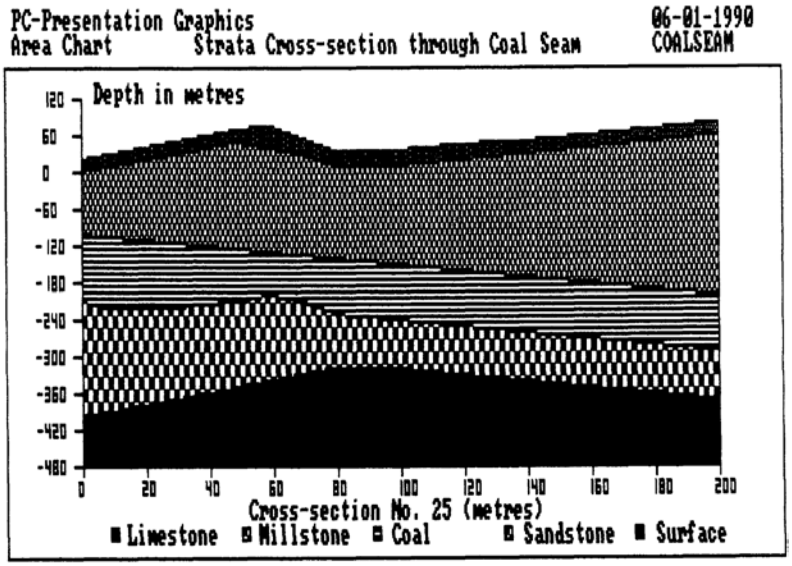

Fig. 2.8 Cumulative area chart: plot of strata cross-section through a coal seam from borehole data. Note use of tiling to differentiate the layers of strata.

$$P(t) = \sum_{i=1}^{m+1} b_i t^{i-1}; \; t_1 <= t <= t_2 \tag{2.13}$$

and

$$P(t) = [x(t)\, y(t)] \qquad (2.14)$$

Thus *P(t)* is a vector of position of the points on the spline. Cubic splines (*m*=3) have proved to be most useful in computer graphics, mainly due to the fact that 3 is the lowest degree of a polynomial that allows the existence of inflection points. The cubic spline can therefore be represented by

$$P(t) = b_1 + b_2t + b_3t^2 + b_4t^3 \qquad (2.15)$$

where b_1, b_2, b_3 and b_4 are unknown constants and the parameter t is in the range t_1 to t_2. The computer listing for this program is given in Appendix B.4. The data entry and display are similar to the previous curves, except it is also possible to include a zigzag line through the data points as well as the cubic spline curve, Fig. 2.7. The data output file with a **'.CUB'** extension lists each of the interpolated cubic spline points between the original data points.

2.3
AREA CHARTS

2.3.1
Cumulative area chart

Appendix B.5

A cumulative area or strata chart is like a multiple line chart with the areas beneath each line shaded, where each data line is measured from the baseline. Note if any of the data lines cross one another, this type of area chart is not suitable and a stacked area chart should be used. The computer listing for area charts is given in Appendix B.5. The initial text screen display is identical to the previous multiple line graph display, the only difference being in the graphics screen display where the area between the graph lines is shaded, in colour if using a colour screen, Plate 1, and by hatching if in monochrome, Fig. 2.8. Note that the boldest or darkest colours should always be on the baseline.

2.3.2
Stacked area chart

Appendix B.5

A stacked area chart shows several data sets, but each area is measured from the last area, thus the areas do not share a common baseline. The screen display is similar to that described for cumulative area charts, Plate 2 and Fig. 2.9.

2.3.3
100% stacked area chart

Appendix B.5

A 100% stacked area chart also shows several data sets, where each area is a percentage of the total of the measurements at that data point. This is

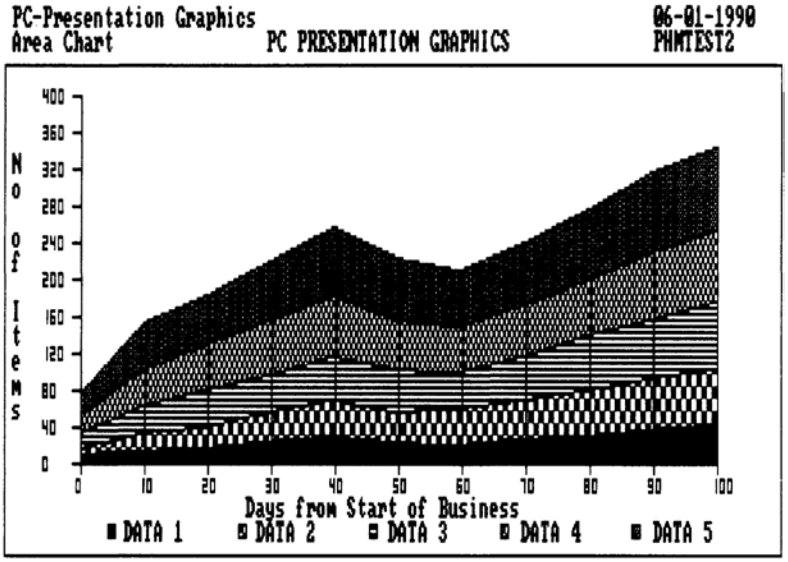

Fig. 2.9 Stacked area chart: each layer of the plot uses the lower layer as its baseline, i.e. no common baseline for reference.

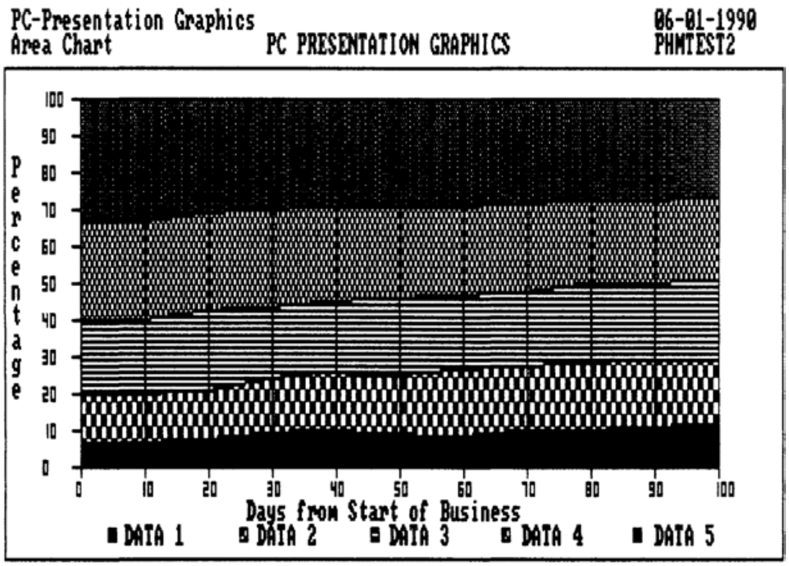

Fig. 2.10 100% stacked area chart: plot shows percentage of total at each data point, useful for comparing percentage trends in data.

useful for comparing percentage trends in data, as shown in Fig. 2.10. Because the areas in the two stacked area charts do not share a common baseline, the order of the data sets (and their corresponding areas) can radically affect the look and meaning of the chart. Where the bands representing each data set are often quite irregularly shaped, it is recommended that the data set with the largest area is placed on the baseline. If a stacked area chart becomes very jagged (extreme swings in data), it is difficult to interpret, and a multiple bar or column chart is recommended as discussed in the next chapter.

3
Bar and column charts

3.1
INTRODUCTION

Bar and column charts are the most common of all the chart types used in presentation graphics. This is a testimony to their ease of creation and versatility, and their general acceptance in company reports. These charts can represent both changes in time or item comparisons in either a vertical (column) or horizontal (bar) format. Vertical columns are best for time series comparisons (similar to line graphs) and horizontal bars are best for item comparisons. The main graphic components of these charts are their rectangular areas whose lengths indicate the data values.

3.1.1
Column charts

Column charts in the vertical format emphasize the values for individual time periods. The height of each column measures the data value for each period. Because viewers naturally associate left-right movement with the movement of time, column charts are better than bar charts to display items that change with time (similar to line graphs). As the column chart is simple and easy to read, it appears less statistical than the standard line graph to persons not too statistically minded. When several data sets are to be compared, either multiple (clustered) or stacked (segmented), column charts can be used. In a multiple column chart, each portion represents a different item for the same time period. Multiple column charts therefore compare related sets of items over time with emphasis on the value of each item in the set, rather than the total sum of the items. If a comparison is required for the totals of the items in each time period, then a stacked column chart is to be preferred, provided the total is meaningful with all items in numerical or monetary terms.

Column charts normally have spaces between the columns or groups of multiple columns, where all the spaces and all the columns should be the same width. However, if the data is for a frequency distribution, and a **histogram** is required, then the spaces can be removed and the emphasis is on the overall area under the histogram rather than on the individual columns.

3.1.2
Bar charts

Bar charts in the horizontal format are best for comparing items at a single point in time. You may need to find the *largest* company, the *highest* profit, the *lowest* interest rate, the *smallest* inventory, and so on. Therefore it is normal to sort or rank the items in numerical order from the smallest to the largest (or vice versa) on the bar chart. Multiple and stacked bar charts can also be used for several data sets. However, it is not possible to rank multiple bar charts, whereas

Table 3.1 Types of bar/column chart

Type of chart	*Description*	*Advantages*	*Disadvantages*
Single bar chart	Heights of bars show frequency of variable. Bars can be vertical or horizontal	Makes comparison easy; clear strong visual impact	Only simple information can be shown
Multiple bar chart	Shows a number of items separately within say a year	Comparing items within say a year and items between the years	Only a few items can be included
Stacked bar chart	Divides an item into its component parts	Shows the division of an item into its constituent parts, while still enabling comparison between total items	Only a few parts of an item can be included
100% stacked bar chart	Bars represent 100%, the components change proportionally	Comparing the constituent parts of the total	Does not show the total output

(Adapted from Honnagan (1986))

it is possible to rank stacked bar charts where the length of the bar indicates the overall total. Bar charts should not be used for time series data.

3.1.3
Types of bar/column chart

The aim of presenting data in a pictorial format is to make an immediate impact and illustrate specific data items. The user therefore has a choice of four different types of bar/column charts as described in Table 3.1, with comments on the advantages and disadvantages of each type. As all the bar/column chart programs use similar routines, they are all included in the computer listing in Appendix C.1, with the split into column or bar charts at Line 550.

3.2
COLUMN CHARTS

3.2.1
Single column chart

Appendix C.1

For many people, the single column chart can be simpler to gauge and more readily understood than a line graph. The single column chart is ideal for time series data showing numerical values for a specific item over a period of time. The frozen moments of time depicted by each column of data therefore receive primary emphasis. It is usual to draw the columns separately to emphasize that each one is a distinct unit, with a space between the columns, Fig. 3.1. If the data are for a statistical frequency distribution, the horizontal scale (X-axis) indicates the characteristic being measured, and

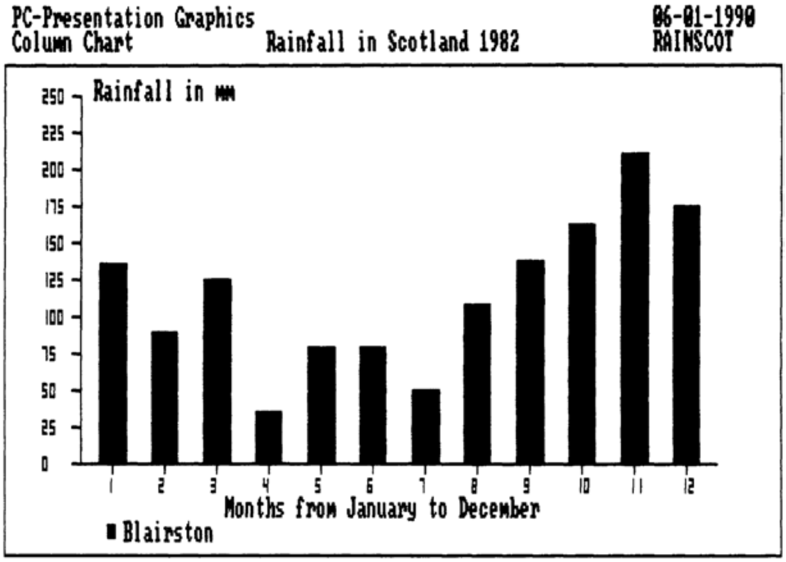

Fig. 3.1 Single column chart: plot of monthly rainfall for Blairston on the Clyde (OS Ref. NS704579) in 1982. (Data for Figs 3.1 to 3.9—extracted from *Hydrological Data UK, 1982 Yearbook,* Institute of Hydrology, Wallingford 1985).

the vertical scale (*Y*-axis) indicates the frequency with which the measured characteristic occurs. In this case, the columns can be joined together to form a histogram, Fig. 3.2. Charts drawn as histograms are useful for analysing cumulative distributions of data—for example, charting income levels among workers, or graphing the most common selling price for a particular type of engineering product.

The format for data presentation and the graphics screen template are identical to that described for line graphs. The computer listing to display a single column chart is presented in Appendix C.1. As before, the program automatically recalls the last data file saved to disc. The user has a choice of loading this data file or specifying another data file. Once loaded, the *X*-axis data cannot be changed, but as before the *Y*-axis maximum and minimum can be altered if desired. If a file of several data sets has been loaded, the user has to choose which data set to display. Finally, the user has to select whether a column chart with spaces or a histogram presentation is required. To compare the height of the columns, a horizontal background grid can be included if desired. On completion of the screen plot, the user can obtain a screen dump by pressing [Shift] and [Prt sc], before re-running the program or returning to the master menu.

3.2.2
Multiple column chart

Appendix C.1

A multiple column chart, sometimes called a clustered, compound or grouped column chart, allows several data sets to be viewed for comparison. Ideally, no more than three or four are recommended as too many data sets become difficult to analyse. Emphasis is on the value of each item at a set time interval, rather than a sum of items. The screen display for the multiple column chart is identical to that for the single column chart, with the addition of a horizontal background grid if desired, Fig. 3.3. If a colour monitor is used, each of the columns in one data set is filled using the **PAINT** command, with the same colour for ease of recognition (e.g. red, magenta, blue, cyan or green), Plate 3. These colours can be changed by the user if desired, by altering the program lines which specify the colour value held in the numerical variable **CL%(J)**. If a monochrome monitor is used, each data set is hatched by a different type of pattern using tiling,

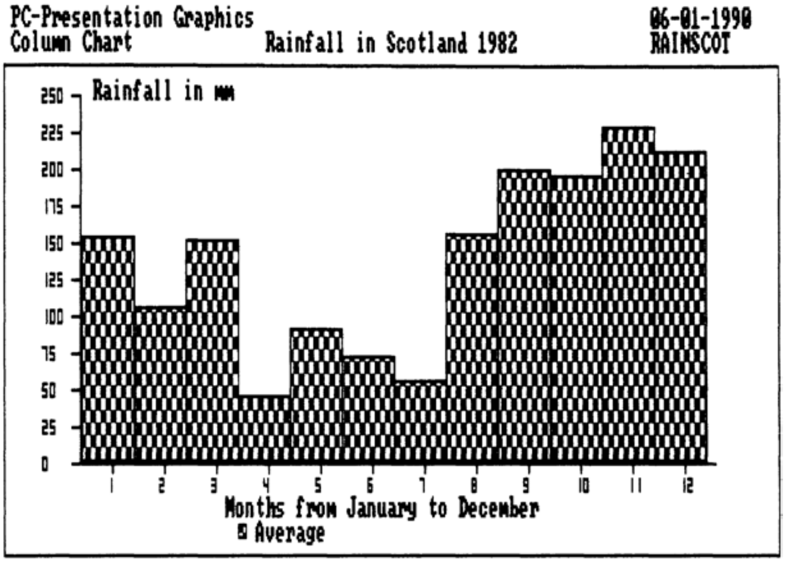

Fig. 3.2 Histogram: plot of average monthly rainfall in Scotland in 1982.

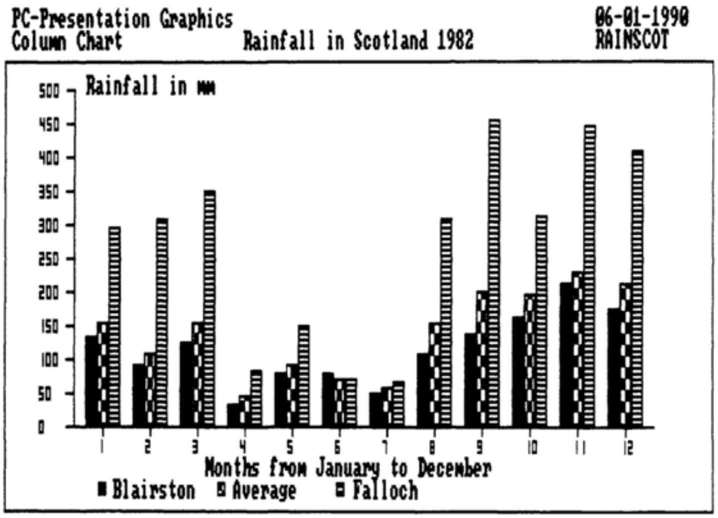

Fig. 3.3 Multiple column chart: study of monthly rainfall in 1982 between Blairston (see Fig. 3.1) and Falloch at Glen Falloch (OS Ref. NN321197), at head of Loch Lomond, compared with the monthly average rainfall.

as described in Section 1.9.13, and the pattern stored in the string variable **TILE$(J)**. The number of sets of multiple columns will depend on the number of rows of data, normally ten to twelve, as too many become confusing. This program utilizes the same general routines as presented in Appendix C.1 for the single column chart.

3.2.3
Stacked column chart

Appendix C.1

A stacked column chart is also often referred to as a component or segmented column chart. The advantage of this type of chart is that a single column, split into segments, is now presented for each time period,

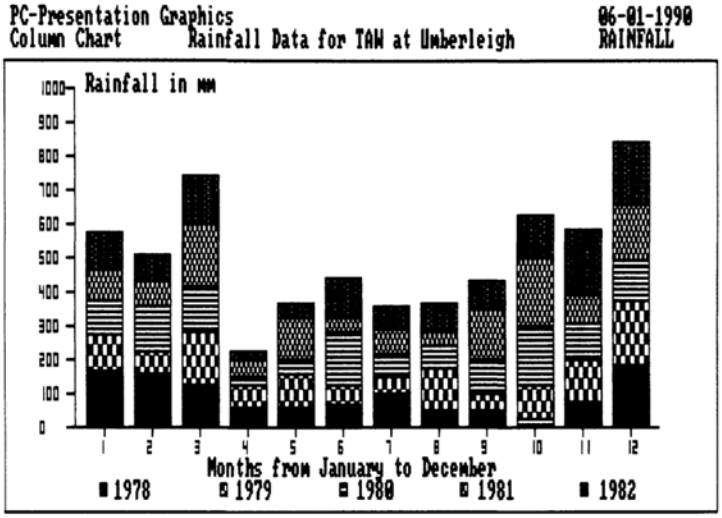

Fig. 3.4 Stacked column chart: plot of monthly rainfall over five years, 1978– 1982, showing cumulative totals over five-year period. Data for TAW at Umberleigh (OS Ref. SS608237), in the area of the South West Water Authority. Note use of five different tiling patterns to differentiate the data sets.

thus giving a comparison of the totals for each of the time periods, Plate 4 and Fig. 3.4. The disadvantage of this type of presentation is that only the lowest segment has a common baseline, thus if comparisons within segments for a given time period are to be analysed, it is better to use a multiple column chart. The text and screen displays are similar to those used in the multiple column chart.

The program listing given in Appendix C.1 includes a special routine at Line 800 to calculate the maximum total for any one column to ensure automatic scaling within the screen display area available. This also adjusts the scaling on the *Y*-axis accordingly.

3.2.4
100% stacked column chart

Appendix C.1

A 100% stacked column chart or **normalized column chart** also shows several data sets, where each segment in the stack is a percentage of the total of the measurements at that period of time. This is useful for comparing percentage trends in the data segments, Fig. 3.5, but does not indicate their overall total value. Although the same information could also be presented in a pie chart (Chapter 4), the 100% stacked column chart offers the distinct advantage of showing several factors proportionally over time in a single chart. The percentages for each segment are calculated in a special routine at Line 900 in the computer listing given in Appendix C.1.

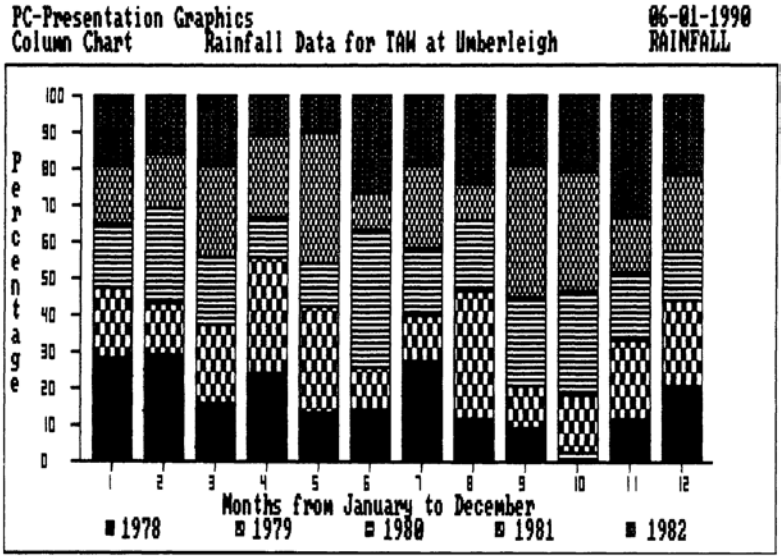

Fig. 3.5 100% stacked column chart: alternative way to plot data of Fig. 3.4 showing the percentage rainfall month by month over a five-year period.

Because the segment areas in the two stacked column charts do not share a common baseline, the order of the data sets (and their corresponding areas) can radically affect the look and meaning of the chart. As mentioned previously, the lowest segment in each column should be bolder than the others and the density of the colour or shading gradually reduced in the higher segments.

3.3
BAR CHARTS

3.3.1
Single bar chart

Appendix C.1

Horizontal bar charts are best for comparing items at a single point in time. An arbitrary, chronological or ranked method of presentation order can be used. It is normal to rank the items *numerically* from the smallest to the largest (or vice versa) on the bar chart, Fig. 3.6. This ranking can affect the viewer's analysis of the chart from optimism (increasing bar lengths) to pessimism (decreasing bar lengths), so the overall message can be emphasized accordingly. An alternative presentation is *alphabetically* by data label, but as the length of the bars will now be irregular, the viewer should use a background grid for comparison.

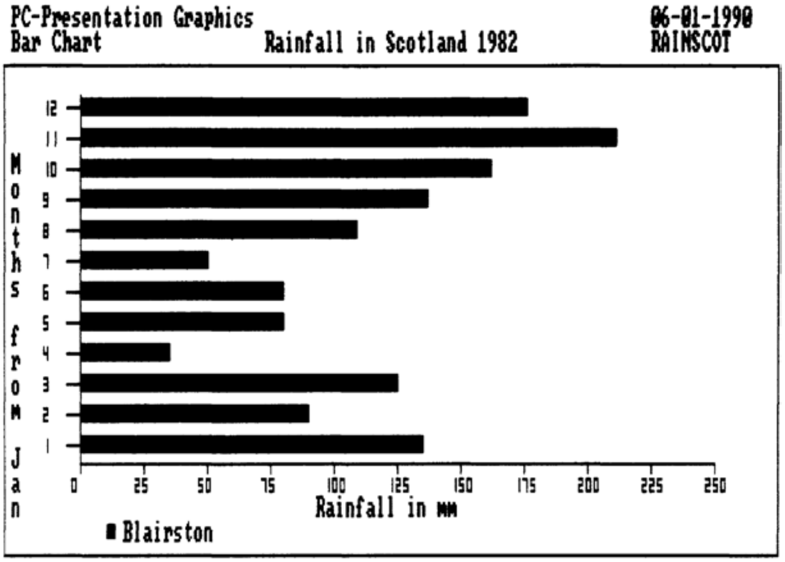

Fig. 3.6 Single bar chart: data of Fig. 3.1 as bar chart.

The initial text and screen presentation is the same as for column charts, until the split at Line 550, where it jumps to Line 2500, as shown in the computer listing given in Appendix C.1. If a vertical background grid is required, it can be added as desired.

3.3.2
Multiple bar chart

Appendix C.1

Horizontal multiple or **clustered bar charts** serve the same basic purpose as vertical multiple column charts. Whereas, three or four data items could be compared in a column chart in the landscape format, only two or three data items should be compared in a bar chart, due to the narrowness of the individual bars. Care should also be taken not to display too many items of data for the same reason given above for the number of data sets. The screen display is similar to that for the single bar chart with the addition of a vertical background grid if desired, Fig. 3.7. As before for column charts, the individual bars will be shaded either in colour or with a pattern for data set identification and comparison. The computer listing for this program is included in Appendix C.1 and uses the same routines as the single bar chart.

3.3.3
Stacked bar chart

Appendix C.1

A stacked bar chart is often referred to as a **component** or **segmented bar chart**. The advantage of this type of chart is that each stacked segment indicates the value of the individual item, while the length of the whole bar shows the total. As before, it is normal to rank the bars in ascending or descending order. It is thus possible to compare individual

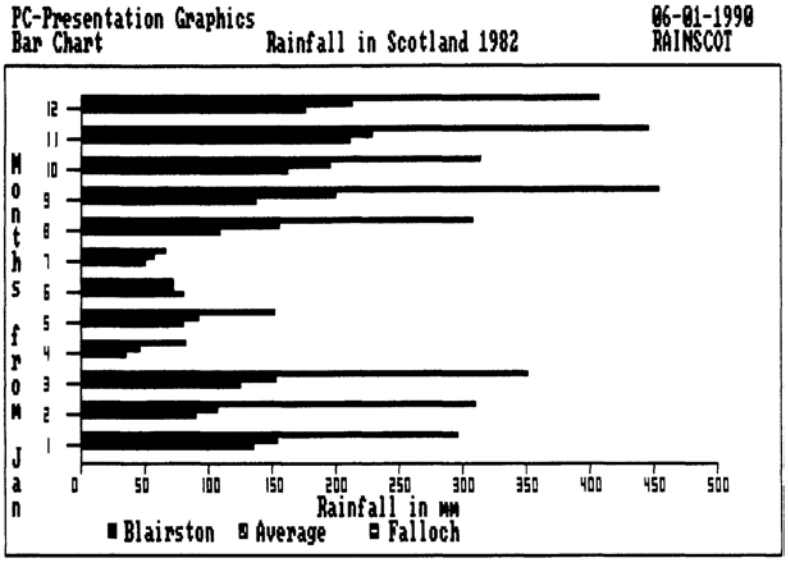

Fig. 3.7 Multiple bar chart: data of Fig. 3.3 as bar chart. Note the loss of tiling when the bars are very narrow.

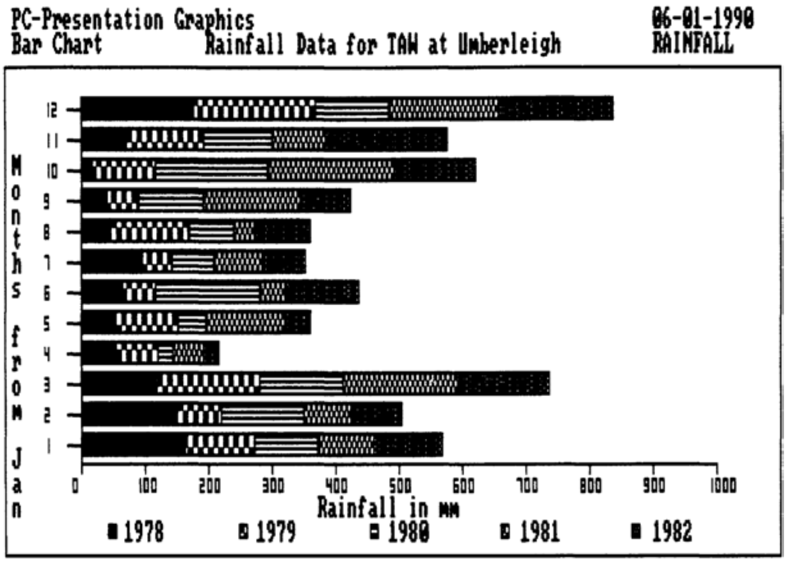

Fig. 3.8 Stacked bar chart: data of Fig. 3.4 as bar chart.

segments between bars and also compare the full length of the bars for the specific data items selected, against a background vertical grid if desired, Fig. 3.8. The disadvantage of this type of presentation is that only the first segment has a common baseline (the vertical axis) and thus it is difficult to make comparisons between segments for different data items. If this type of comparison is required, it is better to use a multiple bar chart. The text and screen displays are similar to that described for the multiple bar chart.

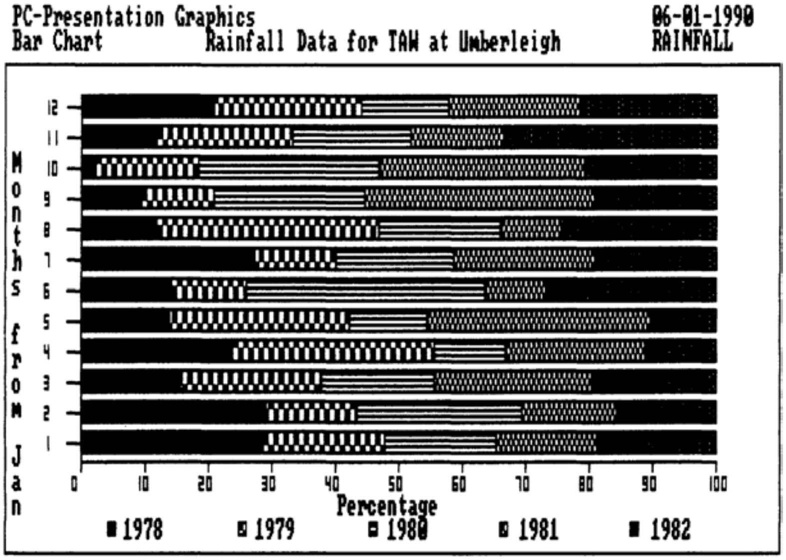

Fig. 3.9 100% stacked bar chart: data of Fig. 3.5 as bar chart.

The program listing given in Appendix C.1 includes a special routine at Line 2800 to calculate the maximum total for any one bar to ensure automatic scaling within the screen display available. This also adjusts the scaling on the *X*-axis accordingly.

3.3.4
100% stacked bar chart

Appendix C.1

A 100% stacked bar chart or **normalized bar chart** also shows several data sets, where each segment in the horizontal bar is a percentage of the total of the measurements for that item. This is useful for comparing percentage trends in the data segments across several data items, Fig. 3.9, but does not indicate their overall value. Although the same information could also be shown in a pie chart (Chapter 4), the 100% stacked bar chart offers the distinct advantage of showing several factors proportionally over several data items in a single chart. The percentages for each segment are calculated in a special routine at Line 2900 in the computer listing given in Appendix C.1. The order of the segment areas and shading shown should follow the same recommendations as for 100% stacked column charts (Section 3.2.4).

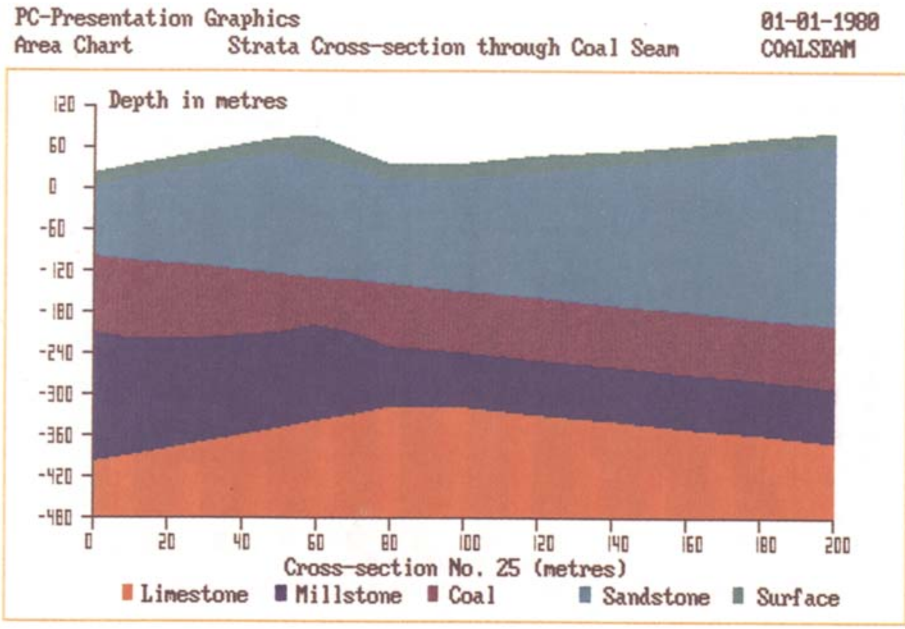

Plate 1 Cumulative area chart: plot of strata cross-section through a coal seam from borehole data. Note use of colour shading to differentiate the layers of strata for comparison with Fig. 2.8. (Section 2.3.1)

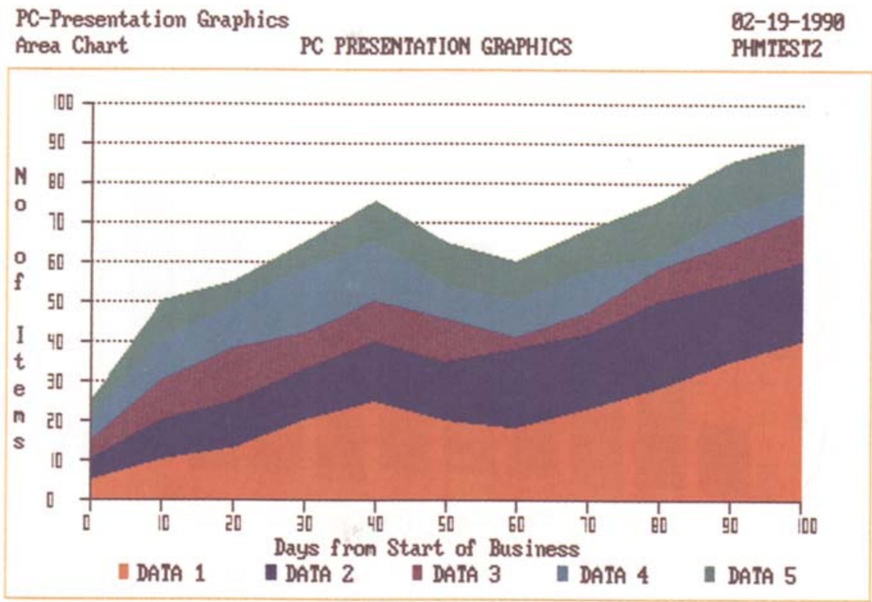

Plate 2 Stacked area chart: each layer of the plot uses the lower layer as its baseline, i.e. no common baseline for reference. Note comparison of colour shading with tiling of Fig. 2.9. (Section 2.3.2)

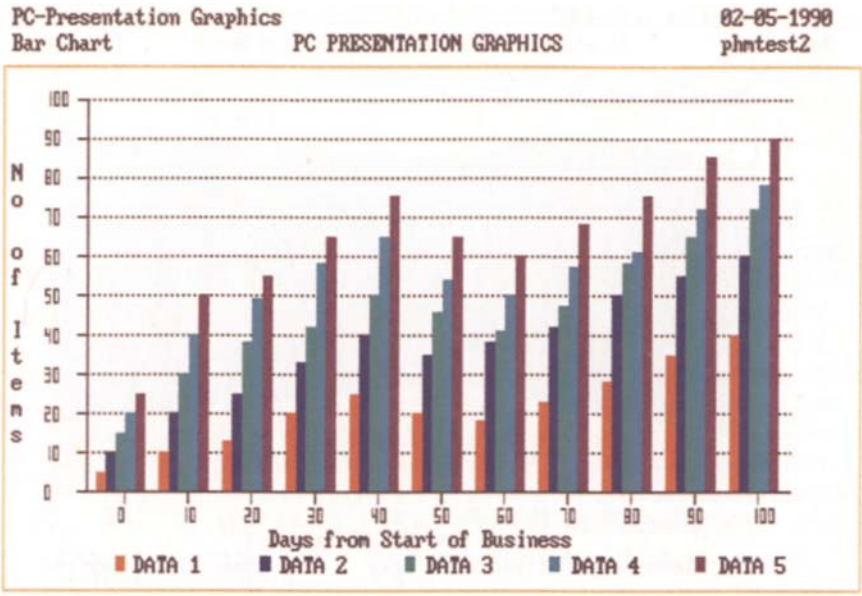

Plate 3 Multiple column chart: colour shading of columns for data of Plate 2 for comparison with tiling in Fig. 3.3. (Section 3.2.2)

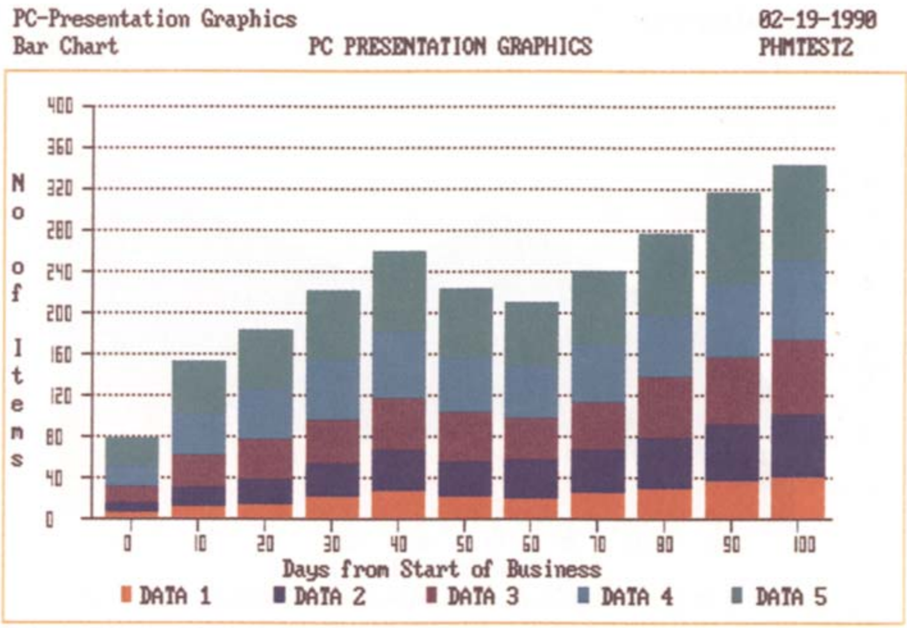

Plate 4 Stacked column chart: colour shading of stacked columns for data of Plate 2 for comparison with tiling in Fig. 3.4. (Section 3.2.2)

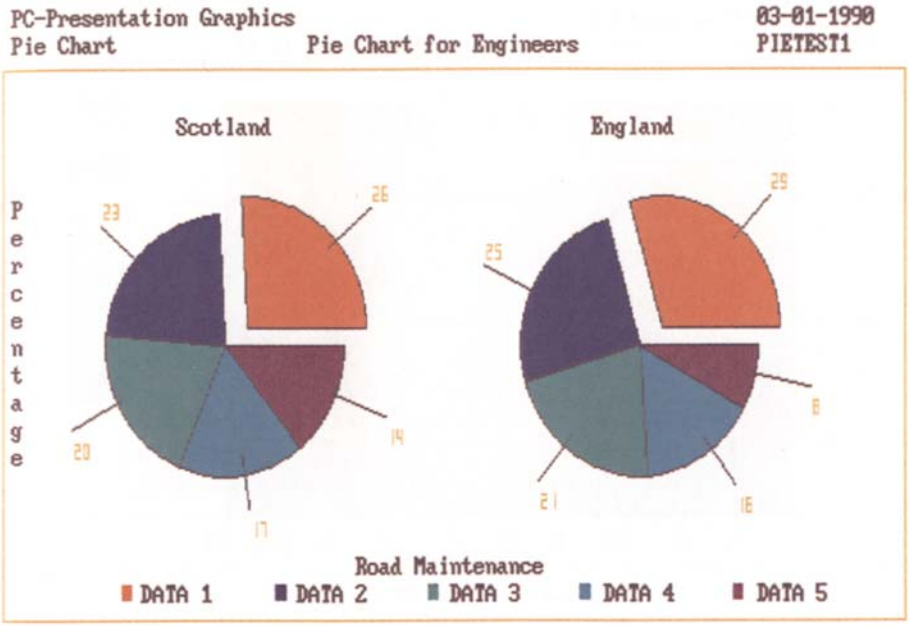

Plate 5 Dual pie chart: results of survey into road maintenance. Note colour shading of pies for comparison with tiling in Fig. 4.3. (Section 4.2.2)

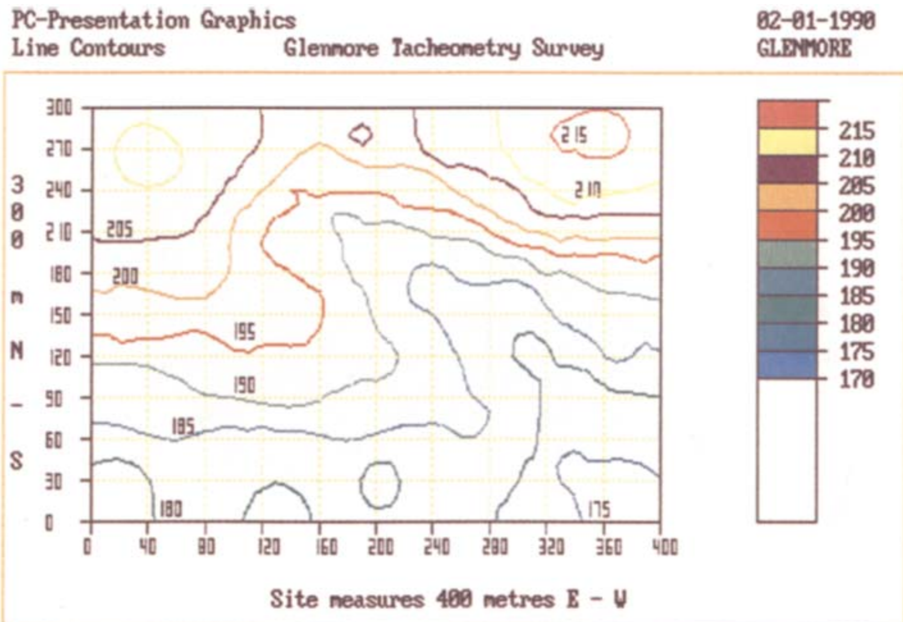

Plate 6 Contour line chart with background grid: screen plot of tacheometry survey at Glenmore, showing site dimensions and contour levels. Note use of 'rainbow' contours for comparison with Fig. 4.5. (Section 4.3.1)

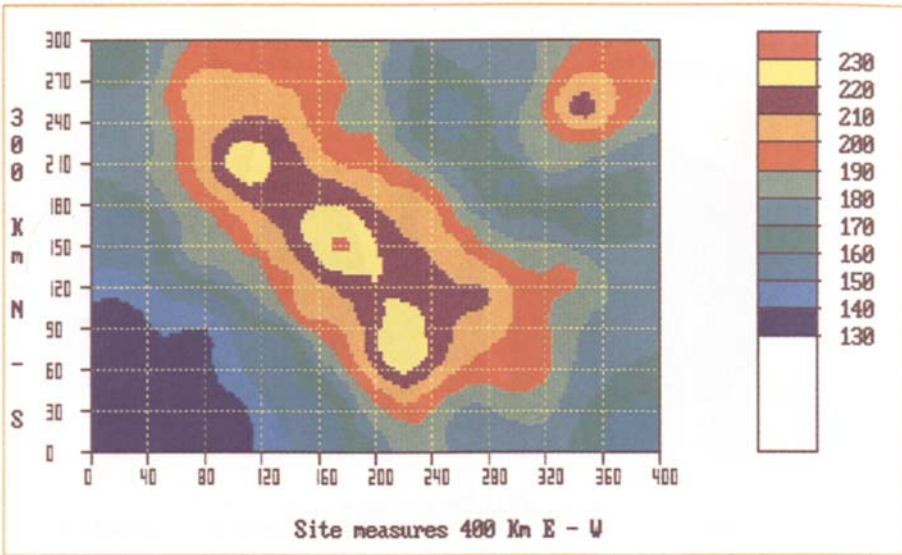

Plate 7 Colour shaded contour chart of rainfall intensity, showing site dimensions and rainfall contours in mm. (Section 4.3.2)

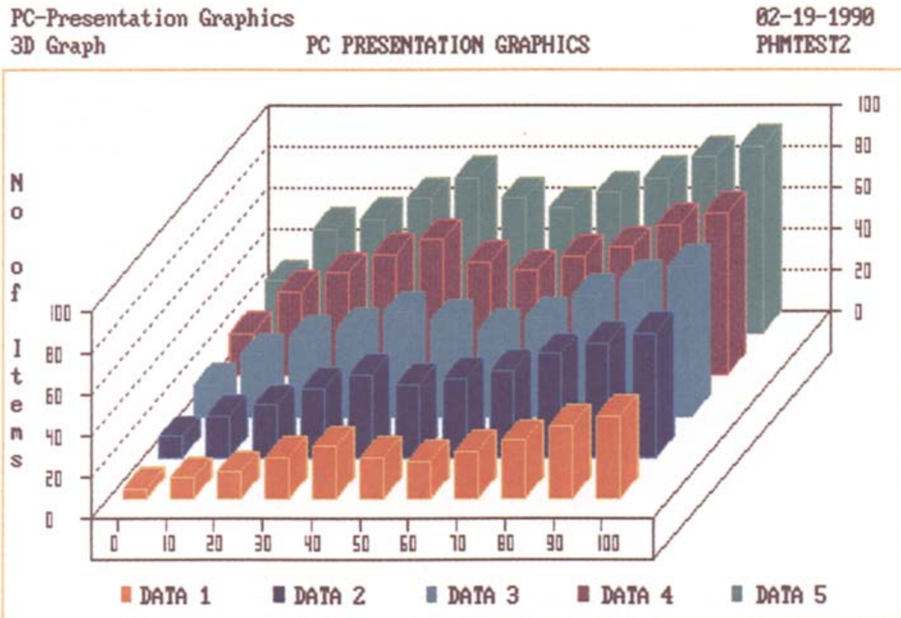

Plate 8 3-D column graph: three-dimensional presentation of five sets of data. Note the need to range the data from the largest items at the back to the smallest in the front. (Section 5.2.1)

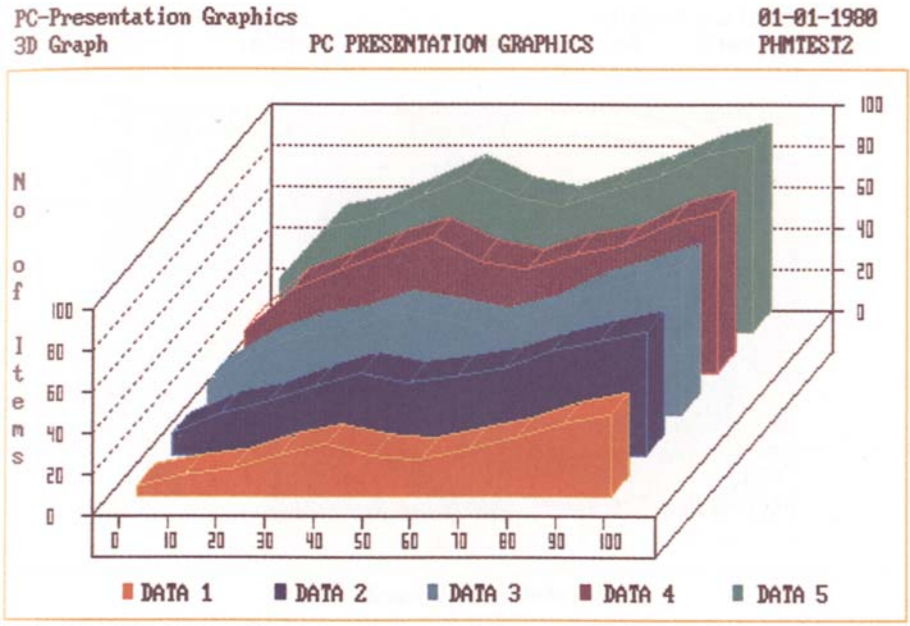

Plate 9 3-D area chart: three-dimensional presentation of data as row oriented ribbon walls with depth—data must be ranged from back to front to ensure data is not obscured. (Section 5.2.2)

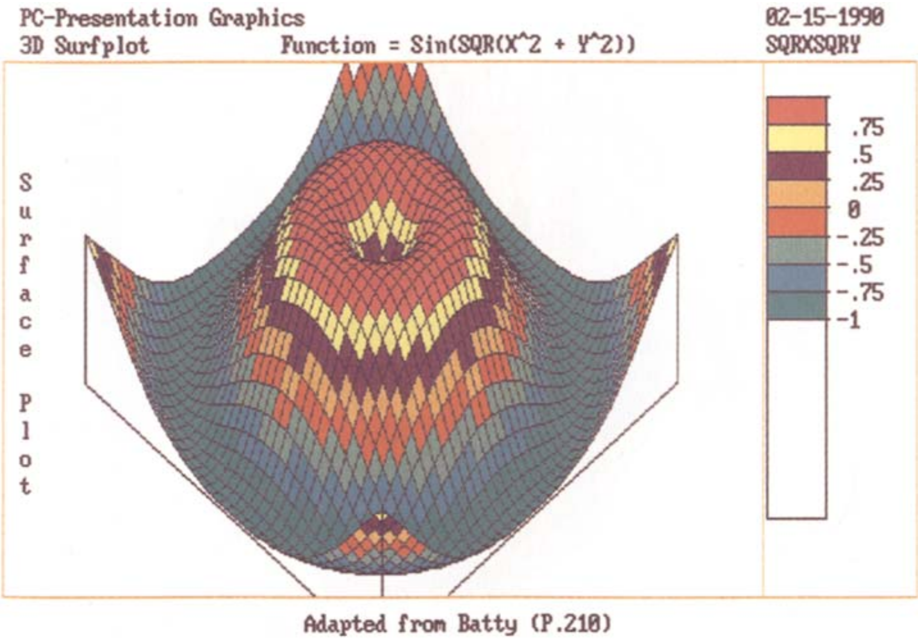

Plate 10 3-D shaded mesh diagram: isometric colour shaded plot of surface function for comparison with Fig. 5.3: equation from Batty (1987). (Section 5.3.4)

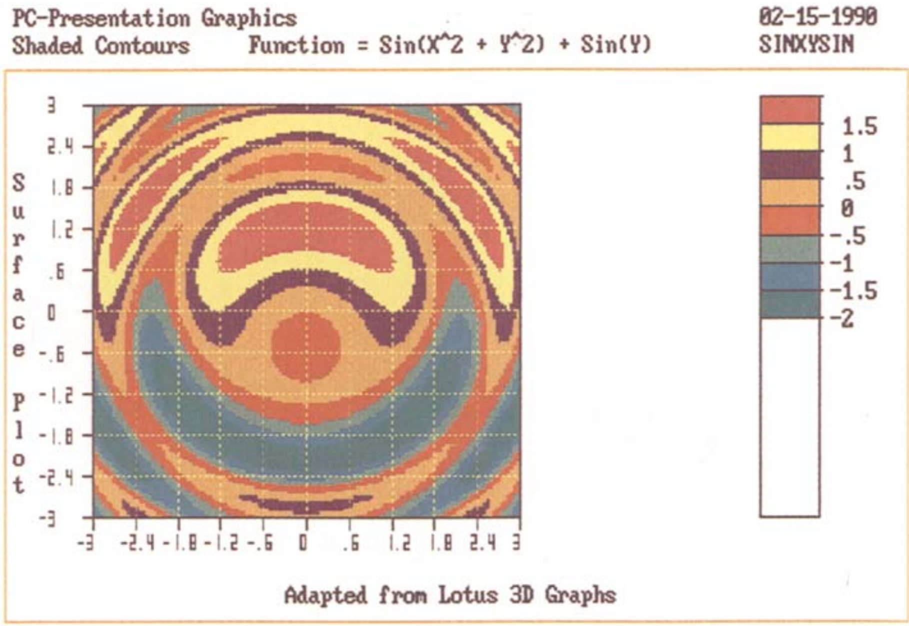

Plate 11 Colour shaded contour chart of Eqt. 1.1 discussed in Section 1.10.3 showing range of *X, Y and Z*-values. (Section 6.2)

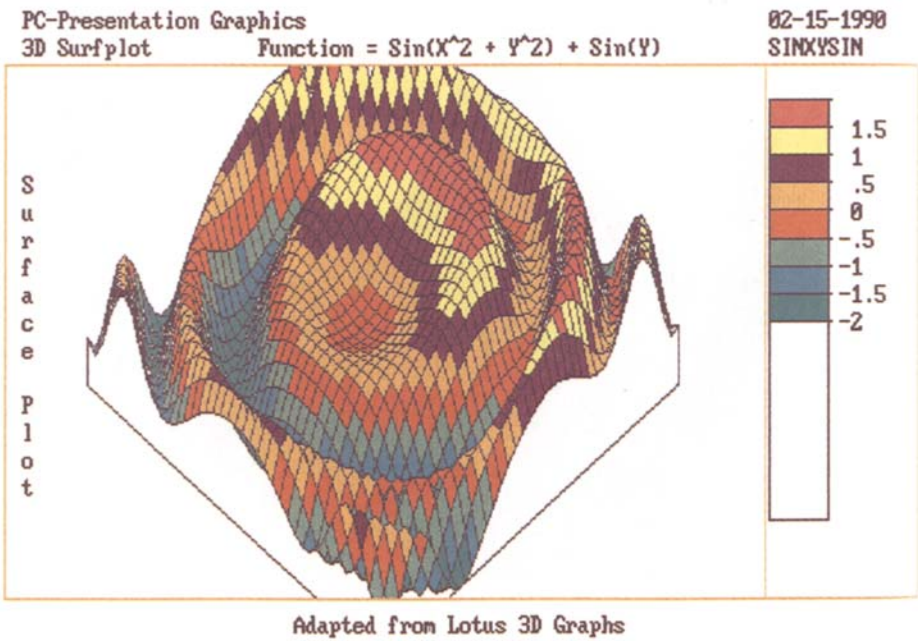

Plate 12 Colour shaded surface model for visualization of contours shown in Plate 11. (Section 6.2)

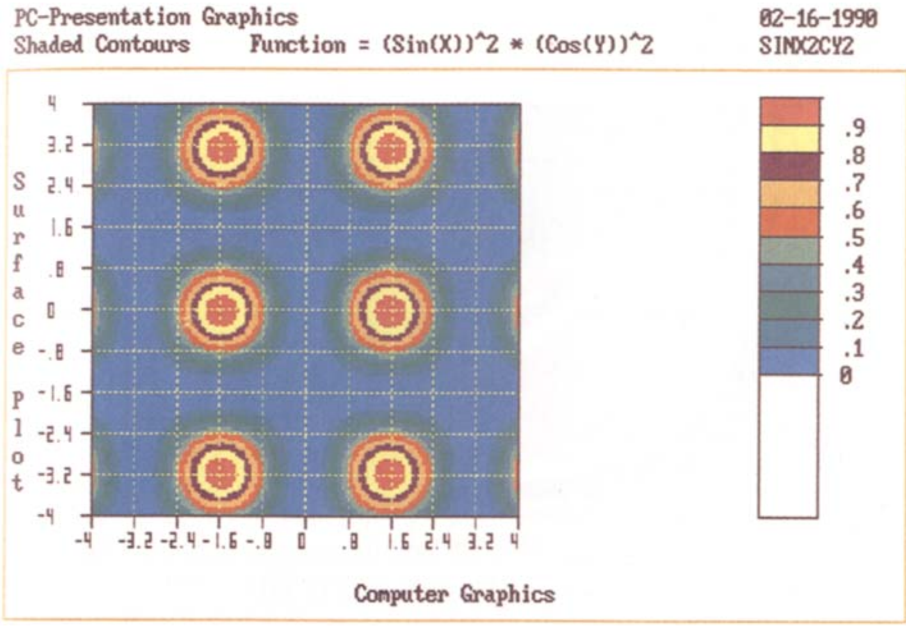

Plate 13 Colour shaded contour chart of $(\mathrm{Sin}(X))^2 \times (\mathrm{Cos}(Y))^2$, showing range of *X, Y and Z*-values. (Section 6.2)

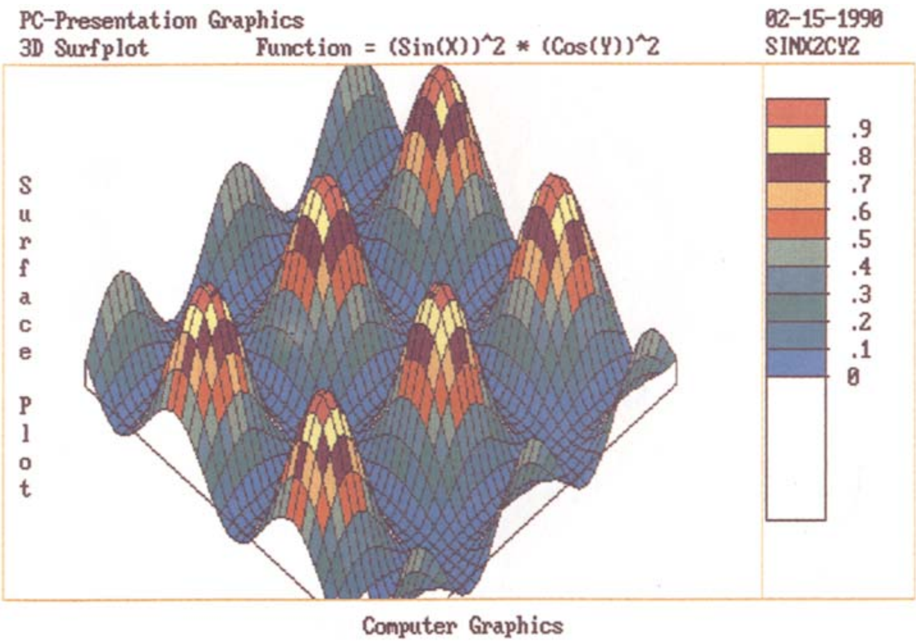

Plate 14 Colour shaded surface model for visualization of contours shown in Plate 13. (Section 6.2)

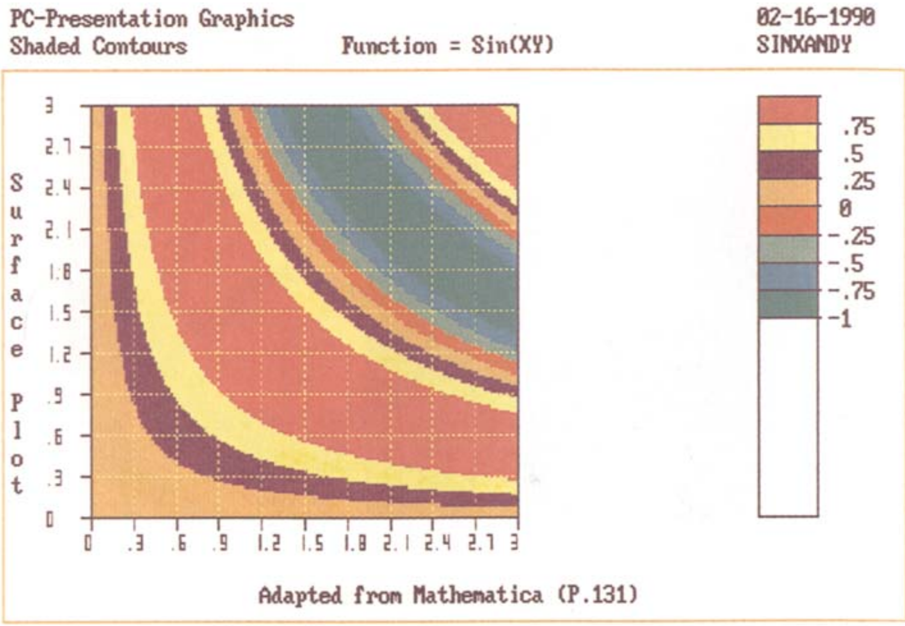

Plate 15 Colour shaded contour chart of Sin *(XY),* showing range of *X, Y and Z*-values: equation from *Mathematica,* (Wolfram, 1988). (Section 6.2)

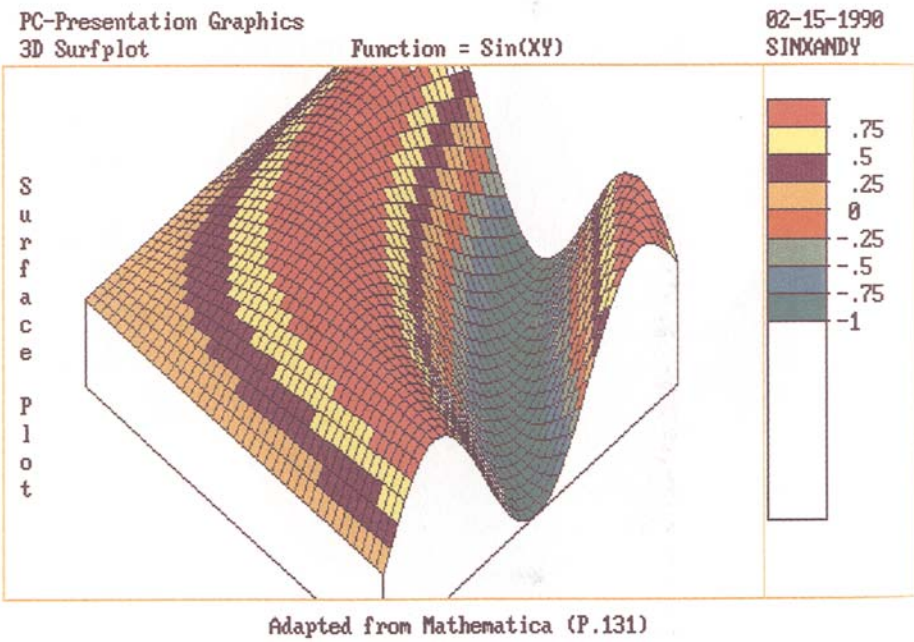

Plate 16 Colour shaded surface model for visualization of contours shown in Plate 15. (Section 6.2)

4
Pie charts and contour maps

4.1
INTRODUCTION

Pie charts are superior to bar charts where four or five comparisons are being made. A pie or circular area chart is essentially a one-dimensional chart that can represent only one set of data per pie, either all items from a single data set or one item from a series of data. Pie charts are therefore suitable for general impressions and for comparisons within a particular class, but are less suitable for a series of class comparisons. Where several comparisons are required, it is possible to use multiple pie charts, but these tend to be confusing, and it is best to limit comparisons to the display of pie charts side by side (dual pie chart).

Contour maps are useful to market researchers, engineers and scientists, who require the two-dimensional representation of three-dimensional data. These can be plotted as line maps or as shaded maps, sometimes called spectral mapped cells.

The choice of chart or map, from the options given in Table 4.1, will depend on the user's requirements and the nature of the data. The advantages and disadvantages of each type of chart are noted in the table.

4.2
PIE CHARTS

4.2.1
Single pie chart

Appendix D.1

In presentation graphics, if the aim of the user is to show how various elements make up a whole, the best form is probably a pie chart. This type of chart is very useful when there are a small number of elements, usually four or five, but never more than eight. Pie charts rely on proportional area to denote quantity and therefore should be appropriate to the data, e.g. area or volume. The standard pie chart represents 100% of a statistic or other item of data. Each of the components of this total is computed as a percentage of the whole 360 degrees and converted to its appropriate segment of the circle. Thus the circle or pie, is divided by radial lines into sectors, so that the area of each sector is proportional to the quantity it represents. Because the pie or circular area chart relies upon immediate impressions, it is not successful when there are consider able variations in the size of the sectors. Pie charts are also not recommended if any of the sectors is less than 5%. For a large number of small data items, it is often better to combine them into one segment under an 'Other' label.

Table 4.1 Types of area chart

Type of chart	*Description*	*Advantages*	*Disadvantages*
Pie chart	A circle divided into sectors to represent each item or variable	Useful where proportions are more important than the numerical values. Can provide a strong visual impact	Can only include a few variables
Dual pie chart	Two circles used for comparison	Useful for comparison	May involve calculations of proportions which do not provide clear visual differences
Line contour map	Contour representation of height data	Plain line contours suitable for monochrome reports	Can only show simple information
Shaded contour map	Spectral mapped cell technique with colour shading to represent height	Strong visual impact in colour where height is important	Can only show relative height information
Area (strata) chart	A chart in layers or stratas	Can show relative importance of items as a whole	Can only include a few variables

(Adapted from Hannagan (1986))

In the presentation of the data, each pie segment should be identified by colour or hatching, and the value or percentage of that segment placed alongside each segment, Fig. 4.1. This labelling is essential, as background grids are not applicable to pie charts for interpolating data values. Where emphasis is required on one specific segment in a data set, that segment should be the first item on the data list so that it can be exploded, as shown in Fig. 4.1. It is also possible in the computer listing given in Appendix D.1 not to

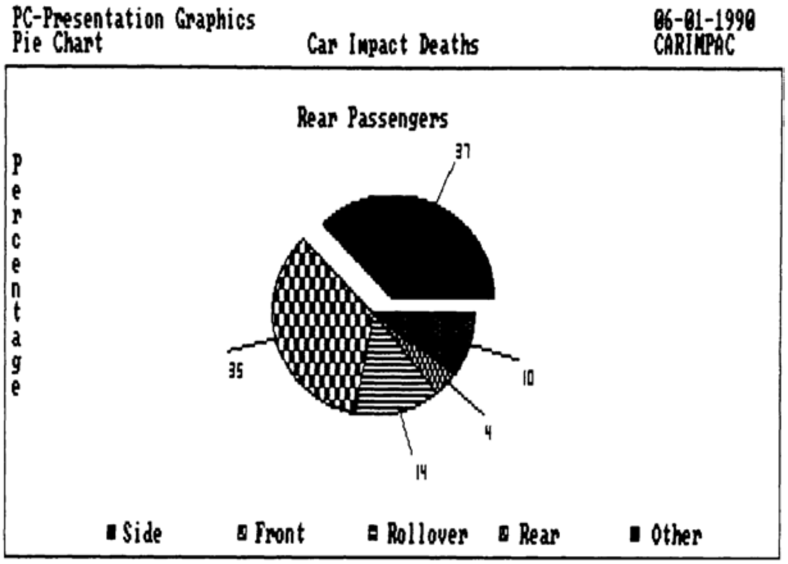

Fig. 4.1 Single pie chart with exploded first segment: *Which?* analysis of UK police accident data identifying the risk of death to rear seat car passengers due to different types of car impact accidents. (Extract of data from *Which?*, October 1989).

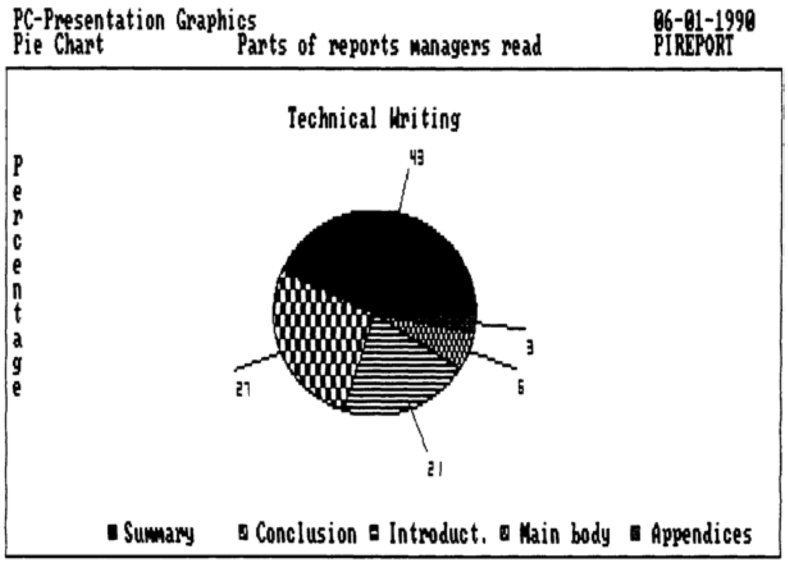

Fig. 4.2 Single pie chart: survey of report reading by 287 managers into the attention paid to the different parts of a technical report. (Extract of data from *Effective Technical Writing and Speaking* by Barry T.Turner).

explode the first segment if required, Fig. 4.2. Segment labels can be by value or percentage as chosen by the user. Data labels to identify the segments are displayed at the foot of the screen.

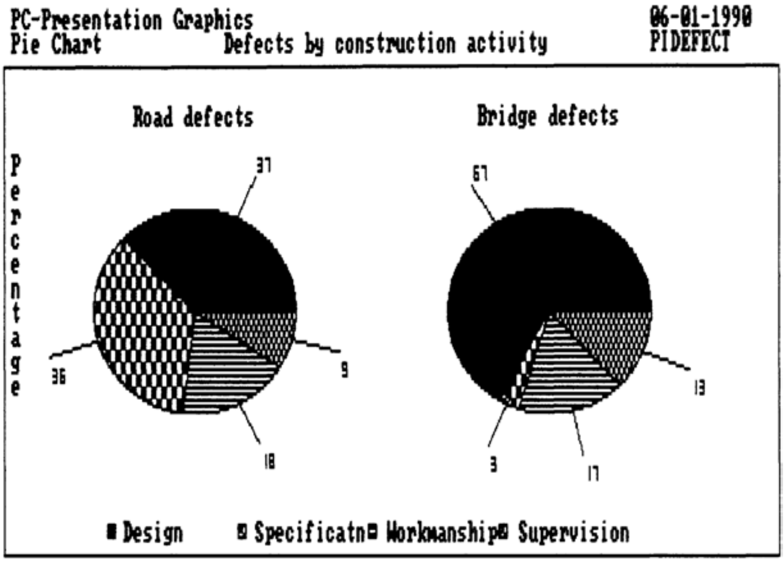

Fig. 4.3 Dual pie chart: results of survey into defects in highway and bridge failures, highlighting design as the major culprit. (Extract of data from *Quality control of road and bridge construction,* National Audit Office, 1989).

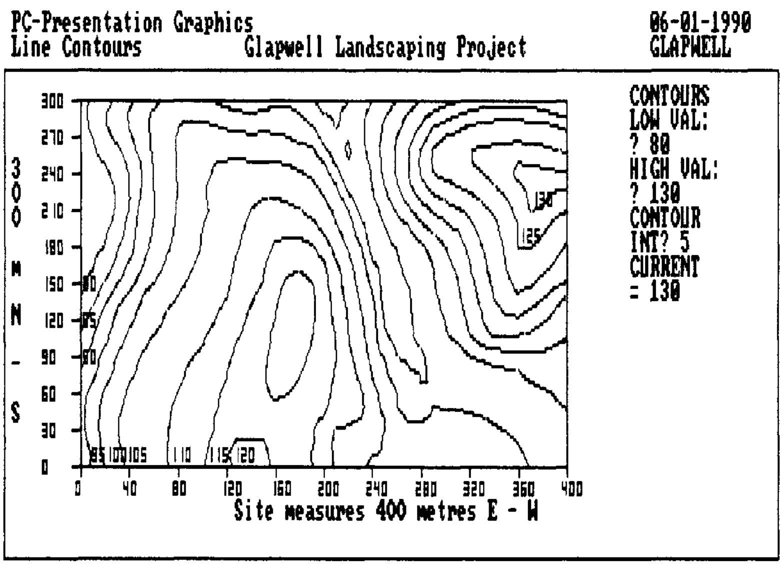

Fig. 4.4 Contour line chart: screen plot of landscaped site at Glapwell showing site dimensions and contour levels.

Data entry for both single and dual pies is by the program **"KEYBDATA",** as discussed in Section 1.10.1 to create both a **'.DAT'** and a **'.TXT'** file. The recalling and displaying of the data is similar to that described for bar charts.

4.2.2
Dual pie chart

Appendix D.1

Dual pie identification labels are entered from the keyboard. The screen display uses the same template for the text, data labels and colours, but without an axis (i.e. **GRAPH=0**), in the computer listing in Appendix D.1. The dual pie chart, as shown in Plate 5 and Fig. 4.3, is very useful for comparing the percentage differences between two sets of data for example, financial data for two different years, or segments between two different data sets, and presents the data more clearly than a multiple bar chart.

4.3
CONTOUR MAPS

Users of Ordnance Survey maps will be familiar with line contour plots to indicate the height of the ground. However, the idea of drawing contours to indicate the shape of the surface is not restricted to land contours, as it can be applied, for example, to stream flow, equipotential values or magnetic or gravimetric intensity. Data to be plotted therefore has to be prepared on a grid basis. Either data has to be sampled on a regular grid and stored in a spreadsheet, or data has to be collected on a random grid and ordered into a regular grid. Such a program, **"PC-RANDOM",** was presented in my previous book, *Computer Graphics for Surveying,* to create a **'.CON'** file containing a regular grid using an inverse square distance weighted interpolation. Such a file can be read by the **"DISCDATA"** program presented in Section 1.10.2. A third alternative is presented in the program **"KEYBEQUD"** where a data file can be generated from any mathematical function, as described in Section 1.10.3. Each of these three methods will produce a data file, commonly called a digital ground model (DGM), which can then be plotted as line contours or shaded contours in 2-D or 3-D.

4.3.1
Line contours

Appendix D.2

The line contour option presented in the computer listing in Appendix D.2 allows the user to select the upper and lower limits for the contours and the contour interval. The maximum and minimum Z values are displayed on the screen for the chosen data file, to guide the user with the contour level selection. Each contour level is displayed on the screen and the line is plotted in sequence from the lowest to the highest level, using different colours with a note of the contour value. A contour following method is adopted, where the lowest contour is traced across the first grid cell in the lower left-hand corner of the screen between the four corner grid levels. Then the next cell to the right is plotted in sequence until all grid cells have been interrogated. After display, the user can add more contours or re-draw the contours at a different contour interval as desired. New users should note that contour levels on any one map should always have the same contour interval. The screen graphics are similar to that used previously for the bar charts, with both the X-axis and Y-scales displayed. The user has also the choice of drawing the contours with or without a background grid for interpolation, etc., as shown in Plate 6 and Figs 4.4 and 4.5.

4.3.2
Shaded contours

Appendix D.3

The shaded contour option presented in the computer listing in Appendix D.3 uses a spectral mapped cell technique to colour code the areas between adjacent contours. A colour shading panel is displayed on the right-hand side of the

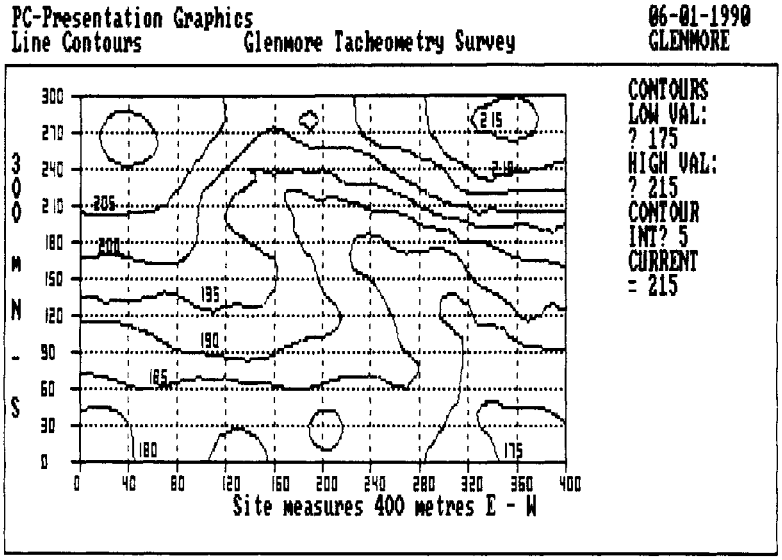

Fig. 4.5 Contour line chart with background grid: screen plot of tacheometry survey at Glenmore, showing site dimensions and contour levels.

screen, giving the colour and associated contour levels. The program will automatically choose the contour intervals, depending on the range of *Z* values to be displayed and the number of screen colours available (16 in EGA). Each grid cell is then scanned and the cell colour shaded accordingly. The program is designed to handle a grid of 50 by 50 cells of data. However, this would give very coarse contour boundaries, so the program further subdivides each original data cell into another 16 cells and interpolates a value for each of these smaller cells before plotting on screen, one cell at a time. Due to the small size of these cells, this program is not suitable for monochrome shading, and will only run on an EGA or VGA colour screen. After display the viewer can obtain a colour screen dump on a range of colour printers, Plate 7. These colours can be changed by the user if required by altering the contents of the colour string **C$** in Line 3260 of the computer listing in Appendix D.3.

5
3-D Charts and surface models

5.1
INTRODUCTION

Although the methods presented in Chapters 2, 3, and 4 will allow you to prepare 2-D charts and graphs for reports and presentations, they are often not as appealing and persuasive as enhanced 3-D charts. The difference between an ordinary and an enhanced chart can be compared to the difference between a sketch and an oil painting, or between a typed draft of an article and the typeset finished version. Straightforward 2-D charts communicate information clearly enough and are fine in reports, but for slide presentations or glossy publications, the audience has a higher expectation and an enhanced professional output is essential.

New graphics programs now enable everyone to create charts which were once the exclusive domain of professional artists. There are a large number of commercial software packages available on the market, and details of some of them are given in Table 5.1. The essence of a good graphics package is the ability to create a template for presentation (Section 1.9.9), with the advantage of enhancement and colour rendering. Each of the programs discussed in this chapter benefit from a colour screen to improve the pictorial effect. The choice of the colours is predetermined by the software routines but can be changed easily. For example, the colour yellow shows up on a black computer screen, but not on a reversed image screen dump, thus the user can alter the colours of background grids as required, i.e. yellow or white grid for colour photographs of the screen, and a blue grid where a colour screen dump is required. Again yellow is not used for the 3-D column or area charts, but is used for the 3-D surface models to give the programs a greater range of colour shading.

5.2
3-D CHARTS

5.2.1
3-D column charts

Appendix E.1

This program will accept the same **'.DAT'** data files as used previously in Chapters 2 and 3. The data recall and display are also similar. To enhance the 3-D perspective, a block base is drawn which includes the *X*-axis markings. Risers or back walls are also drawn with left and right vertical scale labels. The user can choose whether to display a background grid for reference. Colour shading is used to identify each set of data, and data labels are included at the foot of the screen. The graph title and the *Y*-axis label from the **'.TXT'** file are displayed as before, Plate 8. The computer listing for this program is included in Appendix E.1, where the 3-D columns are drawn from back to front to remove hidden

Table 5.1 Presentation graphics packages compared

Program	*Charting*	*Printing*	*Slides*	*Screenshow*
Applause II	Yes	Yes	Yes	Yes
Davrelle	Yes	Yes	Yes	Yes
DrawPerfect	Yes	Yes	No	Yes
Excel	Yes	Yes	No	No
Freelance Plus	Yes	Yes	Yes	Yes
Graph Plus	Yes	Yes	No	No
Harvard Graphics	Yes	Yes	Yes	Yes
Lotus 1-2-3/G	Yes	Yes	No	No
Microsoft Chart	Yes	Yes	Yes	No
Perspective Junior	Yes	Yes	Yes	Yes
Pixie	Yes	Yes	Yes	Yes
Presentation Team	Yes	Yes	No	Yes
Quatro Pro	Yes	Yes	No	No
SuperCalc 5	Yes	Yes	No	No
Storyboard	Yes	Yes	No	Yes
Supergraphics	Yes	Yes	Yes	No
Wingz	Yes	Yes	No	No
Xerox Presents	Yes	Yes	Yes	Yes

lines. Depth shading is used to make the data look three-dimensional. Users should arrange the data so that the series with the largest data values is drawn at the back, with the series containing the smallest data values at the front, so that smaller data series are not obscured. 3-D columns add a dramatic effect to charts, but as they suggest mass and volume, they should only be used for appropriate data.

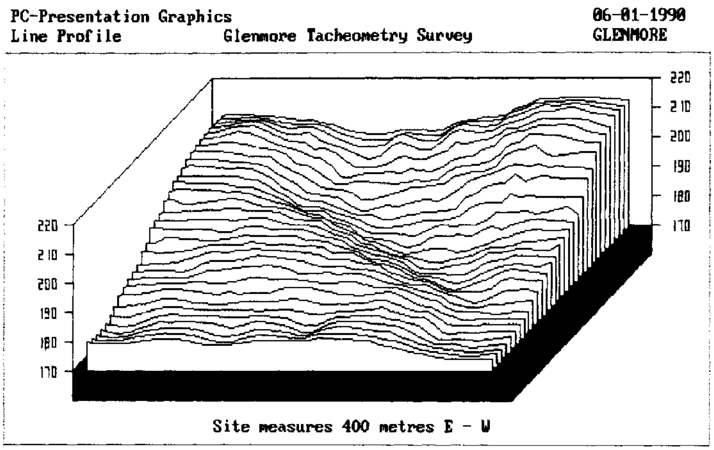

Fig. 5.1 Line profile: plot of tacheometry data at Glenmore (Fig. 4.5) as set of row oriented ribbon walls to emphasize the third dimension—height.

5.2.2
3-D area chart

Appendix E.1

The data display and screen template for 3-D area charts is identical to that described for 3-D column charts. Each set of data is drawn as a set of 3-D area surfaces, with or without a background grid as required, Plate 9. It is most important that the data is ranged from back to front as mentioned under 3-D column charts, to ensure the front set of data does not obscure all the other series.

5.3
3-D SURFACE MODELS

5.3.1
Line profiles

Appendix E.2

It is often useful to present the topography of a site or a frequency distribution as a line profile, sometimes called **row oriented ribbon walls**. The latter refers to the data being plotted row by row from back to front on a 3-D diagram, giving a similar effect to an isometric projection, but without the wireframe effect (which is discussed in Sections 5.3.3 and 5.3.4). The data display and screen template for the line profiles are identical to that described for 3-D column charts, with or without a background grid as required, Fig. 5.1. The computer listing for this program is given in Appendix E.2, where each profile is drawn in a similar fashion to the 3-D area chart program described earlier. The main difference is that the line profile has no depth enhancement due to plotting up to fifty rows compared with five sets of 3-D data.

5.3.2
3-D Line profiles

Appendix E.3

The previous program presented topography as row oriented ribbon walls. This extension to the above program now presents the line profiles as an isometric projection which is useful for visualization of topography, especially in landscape design. The profiles are plotted from the back to the front, and to assist in the viewing of the 3-D perspective, a block base is drawn, covering the plan area of the digital ground model, and showing the vertical height differences at the three visible corners of the plot, Fig. 5.2. The computer listing for this program is given in Appendix E.3, where the initial data screens are similar to the previous programs.

5.3.3
3-D open mesh

Appendix E.4

In addition to producing conventional two-dimensional contour maps (Chapter 4), there is often a wish to display isometric projections (sometimes called **wireframe** or **fish-net** diagrams). It is usual to plot the data using a hidden line removal technique to present a smooth uncluttered view.

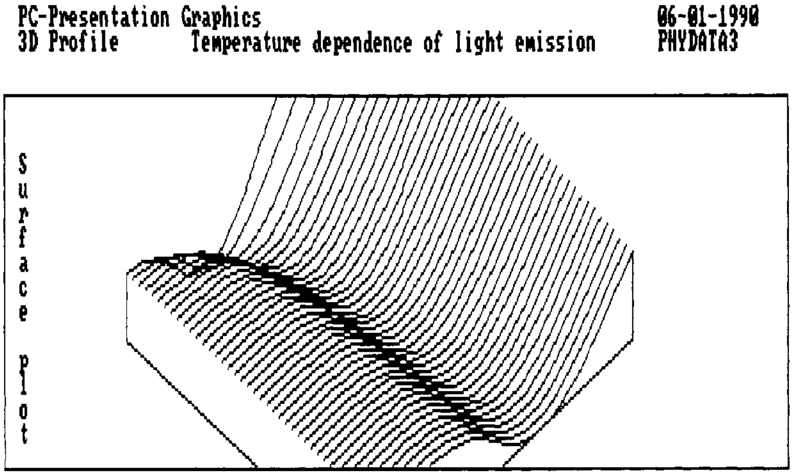

Fig. 5.2 3-D line profile: visualization of surface profile plot of temperature dependence of light emission from a semiconductor sample using iterative numerical methods, to make successive approximations to the best-fit parameter values.

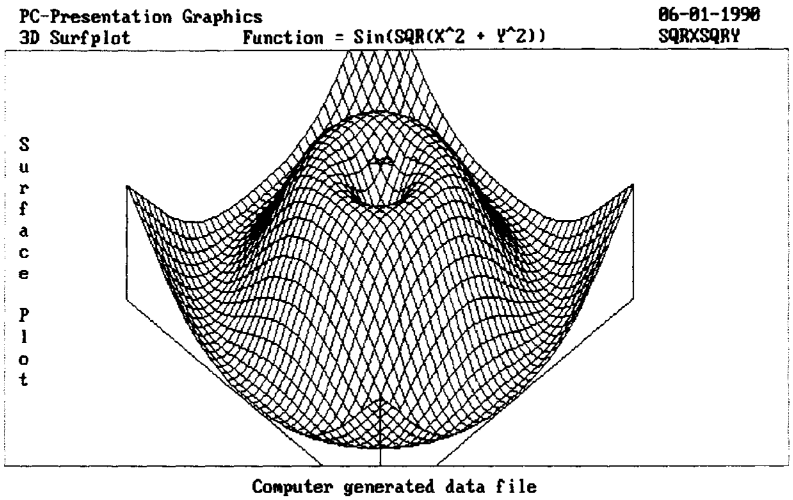

Fig. 5.3 3-D open mesh diagram: isometric plot of surface function, as described in Section 1.10.3: equation from Batty (1987).

Where the wireframe diagram does not show any contours, it is referred to as a 3-D open mesh diagram to differentiate it from a colour contoured 3-D shaded mesh diagram (next section). To assist in the viewing of the 3-D perspective, a block base is drawn, covering the plan area of the digital ground model, and showing the vertical height differences at the three visible corners of the plot, Fig. 5.3. The computer listing for this program is given in Appendix E.4, where the initial data screens are similar to the previous programs.

5.3.4
3-D shaded mesh

Appendix E.5

This program is similar to the 3-D open mesh diagram with the exception that each mesh segment is colour shaded, depending on its average height in the digital ground model. This is sometimes referred to as a **spectral mapped surface** or **shaded mesh diagram**. A colour coded panel is displayed on the right-hand side of the screen for identification, with each of the contour levels marked, Plate 10. This colour shading is identical to that chosen for the shaded contours in Section 4.3.2, and can be changed by the user by altering the contents of the colour string **C$** in Line 4065 of the computer listing in Appendix E.5.

6
Graphics applications

6.1
INTRODUCTION

Graphics is defined in the *Oxford English Dictionary* as 'the use of diagrams as a means of calculation', 'drawing, painting, engraving, etching, etc., graphic representations'. Computer graphics software for business purposes originated in the United States, and its impact can be seen from the large number of software packages available (Table 5.1). However, the engineer or scientist, who wishes to use slides or overhead transparencies for illustration, often has to adapt business software to suit their specific application. The data entry formats are often very complex and require an inordinate amount of time to master a suite of complex key presses often combining the backslash '\' with one-key commands. To simplify this task, the programs in this book have been prepared so that a newcomer to computer graphics should be able to enter data direct from the keyboard in a simple format.

There are therefore several important requirements demanded from a graphics presentation system:

(i) *Easy to use*

This is normally achieved by providing a menu for the user to select the data entry format and the type of presentation and analysis,

(ii) *Economic*

The software should be able to compete with audio-visual slides prepared in the drawing office,

(iii) *Simple to update and change data*

This is an essential feature in scientific disciplines where multiple solutions to problems are often required,

(iv) *Rapid and flexible*

Once data has been entered, the presentation format should be flexible, i.e. the same data can be rapidly viewed by different types of presentation to select the one which is most appropriate.

Once all the programs in this book have been saved to disc, they can be linked together in a presentation graphics menu program, called **"PG-MENU",** as listed in Appendix F.1, where the function keys F1 to F10 are used to select the program required. A second plotter menu program, called **"PLT-MENU"** is listed in Appendix F.2 for those with graphics plotters. Not all the screen graphics programs, which use colour shading, etc., can be adapted to a plotter, and Table 6.1 indicates which programs are available in both formats.

Table 6.1 PC-Presentation Graphics Programs

Chapter	*Description*	*Screen*	*Plotter*
1	Installation	PGINSTAL	–
	Keyboard data	KEYBDATA	–
	Disc data	DISCDATA	–
	Keyboard equation	KEYBEQUD	–
2	Zigzag line graph	GRAPH-B1	GPLOT-B1
	Linear regression	GRAPH-B2	GPLOT-B2
	Exponential curve fit	GRAPH-B3	GPLOT-B2
	Power curve fit	GRAPH-B3	GPLOT-B2
	Cubic spline	GRAPH-B4	GPLOT-B4
	Area chart	GRAPH-B5	GPLOT-B5
	Stacked area chart	GRAPH-B5	GPLOT-B5
	100% stacked area chart	GRAPH-B5	GPLOT-B5
3	Column chart	GRAPH-C1	GPLOT-C1
	Multiple column chart	GRAPH-C1	GPLOT-C1
	Stacked column chart	GRAPH-C1	GPLOT-C1
	100% stacked column chart	GRAPH-C1	GPLOT-C1
	Bar chart	GRAPH-C1	GPLOT-C1
	Multiple bar chart	GRAPH-C1	GPLOT-C1
	Stacked bar chart	GRAPH-C1	GPLOT-C1
	100% stacked bar chart	GRAPH-C1	GPLOT-C1
4	Pie chart	GRAPH-D1	GPLOT-D1
	Dual pie chart	GRAPH-D1	GPLOT-D1
	Line contours	GRAPH-D2	GPLOT-D2
	Shaded contours	GRAPH-D3	–
5	3-D column chart	GRAPH-E1	–
	3-D area chart	GRAPH-E1	–
	Line profile	GRAPH-E2	–
	3-D line profile	GRAPH-E3	GPLOT-E3
	3-D open mesh	GRAPH-E4	GPLOT-E4
	3-D shaded mesh	GRAPH-E5	–
6	Screen menu	PG-MENU	
	Plotter menu	PLT-MENU	

6.2
APPLICATIONS

Presentation graphics give the user an unparalleled opportunity to illustrate routine data in a visual format, with the ability to try different types of presentation before selecting the best display. This book has presented programs suitable for many areas of application in many disciplines, including the following:

borehole logging

contouring

cross-sections

curve fitting

equipotential contours

experimental data

frequency analysis

histograms

least squares

line graphs

linear regression

luminosity

mapping

non-destructive testing

petroleum reservoir analysis

profiling

statistical analysis

stress analysis

surface modelling

terrain modelling

trend graph

3-D visualization

The final choice of graph display will depend on the type of presentation required, either a graph for statistical analysis or a chart/3-D display for visualization. Visualization is one of the most exciting aspects of presentation graphics as it provides a system for the rapid analysis of large quantities of data. Where the sheer volume of data makes it difficult for users to examine more than a fraction of a data set, visualization provides a solution. By allowing large amounts of data to be displayed rapidly on screen, it becomes a powerful tool for analysis and image generation, Plates 11–16.

If it is desired to transfer graphs and charts to either desktop publication packages or slide show systems, the screen displays must be saved as image files or vector plots, as discussed in the next two sections.

6.3
DESKTOP PUBLISHING

Although it is very useful to be able to obtain slides, prints or plots of data, users often require to incorporate their graphic data into reports. With the replacement of the old ‘cut and paste’ techniques with computer based desktop publishing systems, it is essential to be able to save the presentation data on disc in a format acceptable for suitable transfer. Unfortunately there is not a standard format for the transfer of pictorial images. Several formats are available as listed in Table 6.2 with a note on which can be imported by three popular desktop publishing packages.

Table 6.2 Desktop publishing file import formats

Type of file	*Extension*	*PageMaker*	*Ventura Publisher*	*WordPerfect*
Image Files				
Dr. Halo	.PIC	Yes	No	Yes
GEM Paint	.IMG	No	Yes	Yes
MacPaint	.PNT	No	Yes	Yes
MS Windows Paint	.MSP	Yes	No	Yes
PC Paintbrush	.PCX	Yes	Yes	Yes
TIFF Files	.TIF	Yes	Yes	Yes
Line Art				
AutoCAD	.DXF	No	Yes	Yes
	.SLD	No	Yes	No
CGM	.CGM	Yes	Yes	Yes
Encapsulated Postscript	.EPS	Yes	Yes	Yes
GEM Draw	.GEM	No	Yes	No
HPGL	.HPG	Yes	Yes	Yes
Lotus 1-2-3	.PIC	Yes	Yes	No
Macintosh PICT	.PCT	No	Yes	No
Video Show	.PIC	Yes	Yes	No

Each of the computer software packages listed in Table 5.1 provide means of exporting files in one or other of the formats listed in Table 6.2. Care should therefore be taken in the choice of software to ensure compatibility between the export format for the graphics presentation software and the import format for the desktop publishing software.

Where a specific presentation software package does not have the required export format it is possible to purchase a screen capture program. Screen capture utilities 'grab' snapshots of the display on the screen monitor. Once a screen image has been captured, it can then be saved in the required format for transfer to another application, or can be printed on a range of colour printers. Some of the screen capture programs even allow the user to edit or 'clip' the image, thus eliminating extraneous text or command lines. Six popular screen capture programs are listed in Table 6.3 with a note on the export formats available, for reading in conjunction with Table 6.2. Users wishing to use screen capture programs should examine the precise technical specifications for the software they plan to purchase, as occasionally problems can occur with programs running under Microsoft Windows.

Table 6.3 Screen capture programs

Program	*Export formats available*
Collage Plus	.PCX
Grafplus	.PCX
HotShot Graphics	.EPS, .IMG, .PCX, .TIF
Inset and Hijack	.EPS, .HPG, .IMG, .PIC, .PIX
Pinch and Punch	.PIX, .TIF
Pizazz Plus	.EPS, .IMG, .PCX, .SS, .TIF

6.4
SLIDE OR SCREENSHOW

Earlier parts of this book have discussed the preparation of presentation graphics programs on screen, printer and plotter as individual plots. Some software packages also allow screen displays to be saved as individual slide images (see Table 5.1). Two of the main companies involved in producing 35mm slides from screen images are Agfa and Polaroid, and other companies often provide bureaux services.

For graphics presentations to audiences, it often takes too long to demonstrate the data entering and display routines for numerous data files, and as a consequence several software packages allow screen images saved to disc to be displayed sequentially in a screenshow (Table 5.1). The more expensive of these packages will also provide alternative methods of moving from slide to slide at predetermined intervals, that is with fades, wipes, etc. The user should therefore ensure that the slides are not too cluttered with text and information, and are more ‘speaker-supported graphics’ rather than the whole story, which is more fitted to the role of desktop publishing. Any slide show benefits from consistency in the type of presentation, and hence the use of a standard template for the preparation of presentation graphics in this book.

Appendices

INTRODUCTION

Six Appendices are included in this book, giving listings of and printouts from the computer programs referred to in the text. The six are:

Appendix A—*Chapter 1,* general routines and data entry
Appendix B—*Chapter 2,* line graphs and area charts
Appendix C—*Chapter 3,* bar and column charts
Appendix D—*Chapter 4,* pie charts and contour maps
Appendix E—*Chapter 5,* 3-D charts and surface models
Appendix F—*Chapter 6,* menu programs.

These programs were all developed on an IBM PS/2 Model 50 using Microsoft QuickBASIC. The size of each individual program has been limited to approximately 20K bytes to allow the programs to be run on an IBM PC or compatible with 512K memory under Microsoft GW-BASIC, although Microsoft QuickBASIC is preferred for the reasons given in Section 1.3.

The overall strategy of the programs has been to produce straight-forward BASIC programs which could be transferred to other microcomputers. Hence the programming routines have been split into major blocks of 1000 lines, with smaller subroutines starting at multiples of 100 or 500 lines depending on length. Several remarks **(REM)** have also been included to describe each routine. However, if memory space is limited, these could of course be omitted. In the computer printouts, care should be taken to distinguish between the alphabetic letter 'O' and the numerical zero 'Ø' (where Ø is available).

Although BASIC is a useful programming language for scientific calculations, it does not permit subscripts, exponents or the use of italic letters. Thus, where subscripts have been used in the text, these are converted, for example, b_1 is replaced by **B1**. However, exponents, i.e. squares, etc., can be defined, thus x^2 is replaced by **X^2** in BASIC. All lower case parameters in equations, e.g. *x* and *y,* have been changed to capitals, e.g., **X** and **Y**. In normal scientific notation (and text), three types of brackets can be used: '{' '[' and '(', but in BASIC only '(' can be used. Also, where in scientific notation there can be implied multiplication between parameters or parameters and brackets, in BASIC a multiplication sign **'*'** must be specified.

Every reasonable effort has been made during the author's research and testing to ensure the accuracy of the computer programs in this book. The author acknowledges, however, that the possibility of either typo-graphical or

Table A.1 Global numeric variables

A	=	Integer storage of **FACTA** or variable
AA,A1,A2	=	Variables to store pie chart wedge angles
ASPECT	=	Variable to ensure pie chart is circular
AVGX	=	Variable to plot *X*-value
AVGY	=	Variable to plot *Y*-value
B	=	Integer storage of **FACTB** or variable. In Surfplot routine number of pixels in isometric grid (horizontal)
BBY1–3 **BTY1–3** **BX1–3** **BY1–3**	=	Plotter variables used to define 3-D block base
C	=	Loop variable used in pie charts
CC	=	Current contour value
CHG	=	Change variable
CL%(15)	=	Array to identify screen colours
CLOR	=	Colour variable
COLZ	=	Variable for grid column
CX	=	Variable to identify centre of pie chart
CY	=	Variable to identify centre of pie chart
D	=	Variable used in contour interpolation
DATBX	=	Variable to specify type of data identification box
DISP%	=	Variable to select display
DR	=	Variable used in contour interpolation
DS	=	Variable used in colour interpolation
DTOT	=	Data total for pie charts
DUAL	=	Variable to identify dual pie charts
DX	=	*X*-distance of plotted area
DY	=	*Y*-distance of plotted area
ERL	=	Error line number
ERR	=	Error number
FACTA	=	Variable to store line graph value *a*
FACTB	=	Variable to store line graph value *b*
FUNC1	=	Cubic spline function variable
FUNC2	=	Cubic spline function variable
GCOL	=	Graphics screen column variable
GLIN	=	Variable for plotting graphics lines
GRAPH	=	Variable to identify type of graph display required
GROW	=	Graphics screen row variable
GX	=	*X*-coordinate at lower left-hand corner of plot
GX0	=	Screen origin of *X*-axis for grid
GY	=	*Y*-coordinate at lower left-hand corner of plot
GY0	=	Screen origin of *Y*-axis for grid
H	=	Pixel height of graphics numbers
HC	=	Highest contour on plot
HGRID	=	Variable for plotting horizontal grid
I	=	Loop integer

IPXMAX	=	Maximum plotter scaling point coordinate on *X*-axis
IPXMIN	=	Minimum plotter scaling point coordinate on *X*-axis
IPYMAX	=	Maximum plotter scaling point coordinate on *Y*-axis
IPYMIN	=	Minimum plotter scaling point coordinate on *Y*-axis
J	=	Loop integer
K	=	Loop integer
LC	=	Lowest contour on plot
LDAT	=	Variable for plotting *X* and *Y*-axis labels
LN	=	Length of plotting string
LNC	=	Variable used to centre screen titles
LNT	=	Length of screen title for display
LXD	=	Variable for spacing *X*-axis ticks and labels
LYD	=	Variable for spacing *Y*-axis ticks and labels
M(I)	=	Array to store statistical data
MARK	=	Variable for plotting data symbols on graph
MG	=	Multiplication factor for graphics display
MH	=	Multiplication factor for horizontal resolution
MV	=	Multiplication factor for vertical resolution
MX	=	Multiplication to vary pixel spacing
N1–N4	=	Variable to identify value of character in graphics number
NCOL	=	Number of columns of data
ND	=	Number of data sets
NEG	=	Variable to identify negative data values
NOCL	=	Number of screen colours
NP	=	Number of points in contour interpolation
NPOLY	=	Number of polygon vertices
NROW	=	Number of rows of data
NSEL	=	Number of selected item from menu list
NTYP	=	Variable to store type of selection from menu
NX	=	Variable to vary pixel spacing
PGX	=	Plotter variable to replace **GX**
PGY	=	Plotter variable to replace **GY**
PPX	=	Plotter variable to replace **PX**
PPY	=	Plotter variable to replace **PY**
PCX	=	Variable to plot centre of pie chart
PCY	=	Variable to plot centre of pie chart
PI	=	3.1415926
PIE	=	Pie chart variable
PLF%	=	Variable to store type of plotter interface
PLOT%	=	Variable to store size of plotter
PP	=	Screen display variable
PX	=	*X*-coordinate at upper right-hand corner of plot
PY	=	*Y*-coordinate at upper right-hand corner of plot
R1	=	Radius of pie chart
RB	=	Temporary variable to store part of correlation coefficient
RGB%	=	Variable to select colour of lines on screen

RNG	=	Range between minimum and maximum *X*-values
ROWZ	=	Variable for grid row
RT	=	Temporary variable to store part of correlation coefficient
RXY	=	Coefficient of correlation
S	=	Number of pixels in isometric grid (vertical)
SCAX	=	Scaling factor along *X*-axis
SCAY	=	Scaling factor along *Y*-axis
SCD%	=	Screen display selection
SCI%	=	Text or graphics screen selection
SCR%	=	Screen resolution
SDIMN	=	Number of space dimensions
SM	=	Scale factor
SR%	=	Screen resolution variable
SRH%	=	Screen resolution variable
SRX%	=	Screen resolution variable
SRY%	=	Screen resolution variable
ST	=	Variable to store graphics line style
STDX	=	Standard deviation in *X*
STDY	=	Standard deviation in *Y*
SUMLX	=	Sum of log of *X*-values in data set
SUMLXY	=	Sum of product of log of *X* and *Y*-values in data set
SUMLY	=	Sum of log of *Y*-values in data set
SUMLXX	=	Sum of square of log of *X*-values in data set
SUMLYY	=	Sum of square of log of *Y*-values in data set
SUMX	=	Sum of *X*-values in data set
SUMXLY	=	Sum of product of *X*- and log of *Y*-values in data set
SUMXX	=	Sum of square of *X*-values in data set
SUMXY	=	Sum of product of *X*- and *Y*-values in data set
SUMY	=	Sum of *Y*-values in data set
SUMYY	=	Sum of square of *Y*-values in data set
SX	=	Horizontal display width in user units
SY	=	Vertical display width in user units
TCC	=	Global variable to check best spacing interval
TL	=	Variable used in titling contour level
TNL	=	Variable to identify length of graphics number
TSEC	=	Plotter delay variable in seconds
TSEL	=	Selected type of trend line
VEX	=	Vertical exaggeration
VGRID	=	Variable for plotting vertical grid
W	=	Pixel width of graphics numbers
X0	=	*X*-coordinate or screen origin of *X*-axis
X0F	=	Variable used in displaying bar charts
X1–X4	=	*X*-coordinates of points on lines
X3ND	=	*X*-variable used to plot in 3-D
XBAR	=	Mean of *X*-values in data set
XCEN	=	*X*-coordinate of centre of plot

XCND	= Variable to give column spacing on *X*-axis
XM	= Maximum length of *X*-axis
XMAX	= Maximum *X*-value
XMIN	= Minimum *X*-value
XMM	= Maximum *X*-value of plotter sheet size in metres
XND	= Variable to give bar chart spacing on *X*-axis
XP	= Variable to identify *X*-coordinate of graphics number
XSCM	= Temporary variable to store *X*-axis maximum
XSCMAX	= Maximum value of *X*-axis of graph
XTC	= *X*-axis tick spacing on graph
XX1	= Variable to store *X*-values in area charts
XX2	= Variable to store *X*-values in area charts
Y0	= *Y*-coordinate or screen origin of *Y*-axis
Y1–Y4	= *Y*-coordinates of points
Y3ND	= *Y*-variable used to plot in 3-D
YBAR	= Mean of *Y*-values in data set
YCEN	= *Y*-coordinate of centre of plot
YDIST	= Length of *Y*-axis title
YM	= Maximum length of *Y*-axis
YMAX	= Maximum *X*-value
YMIN	= Minimum *Y*-value
YMM	= Maximum *Y*-value of plotter sheet size in metres
YND	= Variable to set bar chart spacings on *Y*-axis
YP	= Variable to identify *Y*-coordinate of graphics number
YP1	= *Y*-coordinate of first point on linear regression line
YP2	= *Y*-coordinate of second point on linear regression line
YPT	= Temporary *Y*-coordinate of point on line
YSCM	= Temporary variable to store *Y*-axis maximum
YSCMAX	= Maximum value of *Y*-axis of graph
YSMIN	= Minimum value of *Y* for screen plot
YT1,YT2	= Temporary storage of *Y*-coordinates
YTC	= *Y*-axis tick spacing on graph
YY1,YY2	= Variables to store *Y*-values in area charts
Z1–Z4	= *Z*-levels of grid points in contour interpolation
Z(I,J)	= Array to store *Z*-values
ZAV	= Average of four grid *Z*-values
ZCUBS	= Number of points on cubic spline
ZD(I,J)	= Temporary array to store *Z*-values
ZG(4,4)	= Array to hold grid *Z*-values
ZMAX	= Maximum *Z*-value
ZMIN	= Minimum *Z*-value
ZP(I,J)	= Temporary array to store *Z*-values
ZT	= Variable used in contour interpolation

Table A.2 Global string variables

A$	=	Answer string
C$	=	String variable to hold screen colours
CAPT$(I)	=	Array to hold pie chart captions
CH$	=	String to store bar chart type
CON$	=	Configuration answer
DAT$(I)	=	Array to store data identifier labels
DATE$	=	Variable which contains computer date
DDSK$	=	Variable to store disc drive location of data files
DESP$	=	Variable to store type of display file
DFILEN$	=	Variable to store name of current data file
DSET$	=	Variable to identify data set to plot
DUP$	=	Variable to display single or dual pie charts
EXT$	=	Variable to store type of data file extension
HGRID$	=	Horizontal grid string
HIST$	=	String to plot as histogram
HPFILE$	=	HPGL plotter file name
K$	=	Variable to store graphics numbers
LABEL$	=	Variable to store display label
LXAXI$	=	Storage for *X*-axis label
LYAXI$	=	Storage for *Y*-axis label
MDSK$	=	Variable to store disc drive location of master files
N1$–N4$	=	Variables to store characters of string **K$**
PC$	=	Storage for type of graph or chart
PCF$	=	Storage for screen and disk specification file name
PLOT$	=	Storage for size of plotter
Q$	=	Response to question
SCD$	=	Storage for type of screen display
SCI$	=	Text or graphics screen
SCR$	=	Storage for screen resolution
SEL$	=	Variable to store menu selection
T$	=	Variable to store software title
TILE$(I)	=	Array to store tiling patterns
TITLE$	=	Storage for title of graph
VGRID$	=	Vertical grid string
Z$	=	Variable to store input string

logical errors exist, and for this reason, those making use of these programs are advised to verify independently the accuracy and correctness of methods and procedures presented in the following pages.

Tables of both numeric variables (Table A.1) and string variables (Table A.2) are included for guidance, as these are common to all the routines and programs. Users should not renumber the routines, which for convenience should be saved for merging into subsequent programs, for example the routine at Line 8000, Appendix A.2 could be saved as **"APP08000.BAS"**.

Appendix A
General routines and data entry

APPENDIX A.1:
GRAPHICS INSTALLATION COMPUTER PROGRAM "PGINSTAL"

This program was discussed in Section 1.7 and allows the user to install a set up file **"PGSCRDSK.PGD"** to check on the type of graphics screen used and the location of the user's program and data files.

Graphics installation

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, the numeric and the string variables which require to be initialized for use in the program. A text screen is specified **(SCREEN 0)** with a standard text width **(WIDTH 40)** to allow the program to be used on any type of screen monitor (TEXT/CGA/EGA/VGA).

(b) Screen display choice Line numbers 100–740

This section of the program allows the user to choose the type of screen monitor used. A graphics screen is required for the display of any of the graphs and charts, but a text screen could be used for the preparation of data. The user can select either a CGA or an EGA screen. If the user has a VGA screen, the EGA option should be chosen since the VGA screen will run in EGA emulation mode, and it is not possible to program the VGA screen from BASIC. The user also has to select whether the presentations should be in colour or monochrome. The user's choices are displayed on screen with the opportunity to change the initial responses if required.

(c) Disc drive choice Line numbers 800–930

The location of both the master presentation programs and the data files have to be entered from the keyboard to ensure the correct running of the programs. Note, if using a hard disc with the data files in a **PGDATA** directory, that two back slashes '\' are required to correctly identify the location, e.g., **'C:\PGDATA\'**. The user's choices are displayed for confirmation or changing before continuing.

(d) Disc installation Line numbers 1000–1110

This section of code saves the graphics installation choices to disk in a **'PGSCRDSK.PGD'** data file. On completion, this program will load the **"PG-MENU"** program for selecting the specific program to be run.
This program also requires the following routine.

(e) Appendix A.15 Line numbers 9000–9190

Graphics installation—BASIC program

```
10 REM <PGINSTAL> Program for PC-Presentation Graphics for IBM-PCs
15 REM ***** Version PC-1.0, 1991 *****
20 REM (C) Copyright 1991 P.H. Milne
25 REM All Rights Reserved
30 REM Title screen at program start
50 CLEAR
60 SCREEN 0
70 WIDTH 40
80 KEY OFF
90 CHG% = 0: SCI% = 0: SCD% = 0: SCR% = 0: SR% = 1: TITLE = 0
100 CLS : PC$ = "Installation"
110 GOSUB 9000
120 PRINT
130 PRINT : PRINT TAB(3); "Select SCREEN"; TAB(20); "<1> TEXT only"
140 PRINT : PRINT TAB(20); "<2> GRAPHICS";
150 SCI$ = INPUT$(1): IF INSTR("12", SCI$) = 0 THEN 150
160 SCI% = VAL(SCI$)
170 IF SCI% = 1 THEN SCR% = 0
180 IF SCI% = 2 THEN 300
190 IF CHG = 1 THEN 500
200 PRINT : PRINT : PRINT TAB(3); "Select DISPLAY";
210 PRINT TAB(20) ; "<1> Monochrome"
220 PRINT : PRINT TAB(20) ; "<2> Colour";
230 SCD$ = INPUT$(1)
240 IF INSTR("12", SCD$) = 0 THEN 230
250 SCD% = VAL(SCD$)
260 IF SCD% = 1 THEN SCD$ = "Mono" ELSE SCD$ = "Colour"
270 GOTO 500
300 PRINT : PRINT : PRINT TAB(3) ; "Select DISPLAY"
310 PRINT : PRINT TAB(6); "<1> Colour -"
320 PRINT TAB(6); "low resolution - CGA"
330 PRINT TAB(6); "<2> Colour -"
340 PRINT TAB(6); "high resolution - EGA"
350 PRINT TAB(6); "<3> Monochrome -"
360 PRINT TAB(6); "low resolution - CGA"
370 PRINT TAB(6); "<4> Monochrome -"
380 PRINT TAB(6); " high resolution - CGA";
390 SCR$ = INPUT$(1)
400 IF INSTR("1234", SCR$) = 0 THEN 390
410 SCR% = VAL(SCR$)
```

```
420 IF SCR% = 1 THEN SCD% = 2 ELSE SCD% = 1
500 CLS
510 GOSUB 9000
520 PRINT : PRINT TAB(3) ; "Installation Details :-"
530 IF SCR% = 0 THEN SCR$ = "TEXT"
540 IF SCR% = 1 THEN SCR$ = "Low resolution"
545 IF SCR% = 1 THEN SCD$ = "Colour - CGA": SCD% = 2
550 IF SCR% = 2 THEN SCR$ = "High resolution"
555 IF SCR% = 2 THEN SCD$ = "Colour - EGA": SCD% = 2
560 IF SCR% = 3 THEN SCR$ = "Low resolution"
565 IF SCR% = 3 THEN SCD$ = "Mono - CGA": SCD% = 1
570 IF SCR% = 4 THEN SCR$ = "High resolution"
575 IF SCR% = 4 THEN SCD$ = "Mono - CGA": SCD% = 1
600 PRINT : PRINT TAB(3) ; "Screen MODE"; TAB(20) ; "- "; SCR$
610 PRINT : PRINT TAB(3) ; "Screen DISPLAY"; TAB(20) ; "- "; SCD$
620 PRINT : PRINT : PRINT TAB(3) ; "Configuration correct (y/n) ";
630 CON$ = INPUT$(1)
640 IF INSTR("YNyn", CON$) = 0 THEN 630
650 IF CON$ = "y" OR CON$ = "Y" THEN 800
660 PRINT : PRINT : PRINT TAB(3) ; "Change <1> MODE, <2> DISPLAY,"
670 PRINT : PRINT TAB(10); "<3> Both, <4> Neither ";
680 CHG$ = INPUT$(1) : IF INSTR("1234", CHG$) = 0 THEN 680
690 CHG = VAL(CHG$)
700 ON CHG GOTO 100, 710, 100, 500
710 IF SCR% = 0 THEN 200
720 CLS
730 GOSUB 9000
740 GOTO 300
800 CLS : GOSUB 9000
810 PRINT : PRINT TAB(3) ; "Select Disc Drives for - "
820 PRINT : PRINT TAB(3) ; "Master Programs (e.g. A: ,C:\PGRAPH\)"
830 PRINT : PRINT TAB(3); : INPUT MDSK$: IF MDSK$ = "" THEN 800
840 PRINT : PRINT : PRINT TAB(3) ; "Saved DATA (e.g. B: ,C:\PGDATA
   \)"
850 PRINT : PRINT TAB(3) ; : INPUT DDSK$: IF DDSK$ = "" THEN 850
860 CLS : GOSUB 9000
870 PRINT : PRINT TAB(3) ; "Installation Details :-"
880 PRINT : PRINT : PRINT TAB(3) ; "Location of Master Programs ";
   MDSK$
890 PRINT : PRINT : PRINT TAB(3) ; "Location of Saved Disc DATA ";
   DDSK$
900 PRINT : PRINT : PRINT TAB(3) ; "Configuration correct (y/n) ";
910 Q$ = INPUT$(1)
920 IF INSTR("YNyn" , Q$) = 0 THEN 910
930 IF Q$ = "Y" OR Q$ = "y" THEN 1000
940 GOTO 800
1000 CLS : GOSUB 9000
1010 PRINT : PRINT : PRINT "Installation data being stored on disc"
1030 FF$ = "PGSCRDSK.PGD"
```

```
1040 OPEN "O", #3, MDSK$ + FF$
1050 PRINT #3, SCR%
1060 PRINT #3, SCD%
1070 PRINT #3, MDSK$
1080 PRINT #3, DDSK$
1090 CLOSE #3
1100 PRINT : PRINT : PRINT : PRINT "PC-Menu now being loaded ....."
1110 RUN MDSK$ + "PG-MENU"
9000 T$ = "Presentation Graphics"
9010 LOCATE 1, 2 * SR%: PRINT T$
9020 LOCATE 1, 31 * SR%: PRINT DATE$
9030 LOCATE 2, 2 * SR%: PRINT PC$
9040 LOCATE 2, 31 * SR%: PRINT TIME$
9050 LOCATE 3, 1: FOR K = 1 TO 40 * SR%: PRINT "="; : NEXT K
9060 PRINT
9070 IF TITLE = 1 THEN 9090
9080 GOTO 9100
9090 RETURN
9095 REM ***** PC-Graphics Title Page *****
9100 FOR I = 6 TO 20
9110 FOR J = 1 TO SR%
9120 LOCATE I, I - 4: PRINT "PC-Presentation Graphics"
9130 NEXT J
9140 NEXT I
9150 LOCATE 23, 5 * SR%: PRINT "Copyright (C) 1991 by P.H.Milne"
9160 TITLE = 1
9170 START = TIMER
9180 IF (TIMER - START) > 3 THEN 9190 ELSE 9180
9190 CLS : GOTO 9000
```

APPENDIX A.2:
GRAPHICS SCREEN AND DISC SET UP COMPUTER ROUTINE AT LINE 8000

This routine was discussed in Section 1.9.1 and ensures that the program runs correctly on the type of screen used (CGA or EGA), and determines whether colour commands are to be used for colour shading or monochrome hatching. The file name **'PGSCRDSK.PGD'** is stored in the string variable **PCF$** before using this routine which is called by using the command **GOSUB 8000** and hence the **RETURN** statement on the last line. This routine also determines the location of the master programs **MDSK$** and the location of the data files **DDSK$**.

Check on graphics screen and disc set up—BASIC routine

```
8000 OPEN "I", #3, PCF$
8010 INPUT #3, SCR%
8020 INPUT #3, SCD%
8030 INPUT #3, MDSK$
8040 INPUT #3, DDSK$
8050 CLOSE #3
8060 ON (SCR% + 1) GOTO 8070, 8080, 8090, 8100, 8110
```

```
8070 SCREEN 0: WIDTH 40: GOTO 8170
8080 SCREEN 2: GOTO 8120
8090 SCREEN 9: GOTO 8120
8100 SCREEN 2: GOTO 8120
8110 SCREEN 2: GOTO 8120
8120 IF SCD% = 1 THEN 8170
8130 IF SCR% = 1 THEN 8170
8140 IF SCR% = 2 THEN 8160
8150 IF SCR% > 2 THEN 8170
8160 COLOR 15, 1
8170 KEY OFF: CLOSE
8180 RETURN
```

APPENDIX A.3:
RECALL OF DATA FILE COMPUTER ROUTINE AT LINE 8200

This routine was discussed in Section 1.9.2 and opens the data file **'IDFILE.PGD',** which contains the name of the last data file, either entered from the keyboard or loaded from disc. This file holds the name of the data file **'DFILEN$'** and its extension **'.EXT$',** which identifies the type of file, either **'.DAT'** for graph data, **'.CON'** for contour data or **'.EQD'** for a generated surface equation. This routine is called by issuing the command **GOSUB 8200** and hence the **RETURN** statement on the last line.

If the data file contained in **'IDFILE.PGD'** is not to be loaded, a second segment of code stored at Line 8800 is called from Line 8280. Once the new data file has been identified the program resumes at Line 8290.

Recall of data file—BASIC routine

```
8200 CLS : GOSUB 9000
8210 LOCATE 6, 10: PRINT "Please wait ..... finding File - "
8220 OPEN "I", #2, DDSK$ + "IDFILE.PGD"
8230 INPUT #2, DFILEN$
8240 INPUT #2, EXT$: CLOSE #2
8250 LOCATE 6, 45: PRINT DFILEN$
8260 LOCATE 8, 10: PRINT "Load DATA for above File (Y/N)";
8265 A$ = INPUT$(1)
8270 IF INSTR("YNyn", A$) = 0 THEN 8260
8280 IF A$ = "N" OR A$ = "n" THEN 8800
8290 LOCATE 10, 10: PRINT "Please wait ..... reading DATA"
8300 OPEN "I", #1, DDSK$ + DFILEN$ + EXT$
8310 INPUT #1, NROW, NCOL
8320 ND = NCOL - 1
8330 FOR I = 1 TO NROW
8340 FOR J = 1 TO NCOL
8350 INPUT #1, Z(I, J)
8360 NEXT J
8370 NEXT I
8380 CLOSE #1
8390 RETURN
```

```
8800 LOCATE 10, 10: PRINT "Select Extension"
8810 LOCATE 10, 30: PRINT "<1> '.DAT' Data File"
8820 LOCATE 12, 30: PRINT "<2> Other Extension"
8830 LOCATE 14, 30: PRINT "<3> Return to Menu "; : A$ = INPUT$(1)
8840 IF INSTR("123", A$) = 0 THEN 8830
8850 NSEL = VAL(A$)
8860 ON NSEL GOTO 8870, 8880, 9600
8870 EXT$ = ".DAT": GOTO 8900
8880 LOCATE 18, 10: PRINT "Enter Extension "; : INPUT EXT$
8900 CLS : DFILEN$ = " ": GOSUB 9000
8905 PRINT : FILES DDSK$ + "*" + EXT$
8910 LOCATE 15, 5: PRINT "Type New Model File Name"
8920 LOCATE 17, 5: INPUT "(Type 'QUIT' to Return to Menu)"; DFILEN$
8930 IF DFILEN$ = "QUIT" OR DFILEN$ = "quit" THEN 9600
8940 OPEN "O", #2, DDSK$ + "IDFILE.PGD"
8950 PRINT #2, DFILEN$
8960 PRINT #2, EXT$
8970 CLOSE #2
8980 CLS : GOSUB 9000
8990 GOTO 8290
```

APPENDIX A.4:
WRITING OF TEXT FILE TO DISC COMPUTER ROUTINE AT LINE 8400

This routine was described in Section 1.9.3 and writes a data file with the extension **'.TXT'** to store the graph title **(TITLE$),** the *X*-axis label **(LXAXI$),** the *Y*-axis label **(LYAXI$)** and its orientation **(YLAB$),** either horizontal **(H)** or vertical **(V),** and the data series labels **DAT$(I)** which are stored in an array. This routine is called by issuing the command **GOSUB 8400** and hence the **RETURN** statement on the last line.

Writing of text file to disc—BASIC routine

```
8400 OPEN "O", #1, DDSK$ + DFILEN$ + ".TXT"
8410 PRINT #1, TITLE$
8420 PRINT #1, LXAXI$
8430 PRINT #1, LYAXI$
8440 PRINT #1, YLAB$
8450 FOR I = 1 TO NCOL - 1
8460 PRINT #1, DAT$(I)
8470 NEXT I
8480 CLOSE #1
8490 RETURN
```

APPENDIX A.5:
RECALL OF TEXT FILE FROM DISC COMPUTER ROUTINE AT LINE 8500

This routine was described in Section 1.9.4 and recalls the text file **'.TXT'** saved to disc in Appendix A.4 where all the string variables are identical. This routine is called by issuing the command **GOSUB 8500** and hence the **RETURN** statement on the last line. Note that if this routine is used and no text file has previously been saved, an error trap must be provided in Appendix A.17 (Section 1.9.16), either to ignore the file or to enter the data before displaying the graph, as described in Appendix B.1.

Recall of text file from disc—BASIC routine

```
8500 OPEN "I", #1, DDSK$ + DFILEN$ + ".TXT": REM Error Line
8510 INPUT #1, TITLE$
8520 INPUT #1, LXAXI$
8530 INPUT #1, LYAXI$
8540 INPUT #1, YLAB$
8550 FOR I = 1 TO NCOL - 1
8560 INPUT #1, DAT$(I)
8570 NEXT I
8580 CLOSE #1
8590 RETURN
```

APPENDIX A.6:
DATA PRINTOUT ROUTINE COMPUTER ROUTINE AT LINE 6000

This routine was described in Section 1.9.5 and allows the user to obtain a printout of the data file **'.DAT'** and the associated text file **'.TXT'** for reference. This routine is called by issuing a **GOTO 6000** command. The last line can either return the user to the program or exit, as shown, to the end of the program at Line 9600.

Data printout routine—BASIC routine

```
6000 LPRINT T$
6010 LPRINT PC$
6020 LPRINT "DATA File - "; DFILEN$
6030 LPRINT "**********************"
6040 LPRINT : LPRINT TAB(5) ; "X-AXIS";
6050 FOR D = 1 TO ND - 1: LPRINT TAB(5 + 10 * D) ; "DATA"; D;
6060 NEXT D
6070 LPRINT TAB(5 + 10 * ND) ; "DATA" ; ND
6080 LPRINT "==========";
6090 FOR P = 1 TO ND - 1: LPRINT TAB(2 + 10 * p) ; "=========";
6100 NEXT P
6110 LPRINT TAB(2 + 10 * ND); "========="
6120 FOR I = 1 TO NROW
6130 FOR J = 1 TO NCOL - 1
6140 LPRINT USING "######.###"; Z(I, J);
6150 NEXT J
6160 LPRINT USING "######.###"; Z(I, NCOL)
6170 NEXT I
6180 LPRINT "==========";
6190 FOR P = 1 TO ND - 1: LPRINT TAB(2 + 10 * P) ; "========="
6200 NEXT P
6210 LPRINT TAB(2 + 10 * ND); "========="
6220 LPRINT
6230 LPRINT "Graph Title - "; TITLE$
```

```
6240 LPRINT "X-AXIS Title - "; LXAXI$
6250 LPRINT "Y-AXIS Title - "; LYAXI$
6260 LPRINT : LPRINT "DATA Legend"
6270 LPRINT "=============="
6280 FOR I = 1 TO NCOL - 1
6290 LPRINT I ; TAB(5) ; DAT$(I)
6300 NEXT I
6310 LPRINT "=============="
6320 GOTO 9600
```

APPENDIX A.7:

DATA ANALYSIS PRINTOUT ROUTINE COMPUTER ROUTINE AT LINE 6500

This routine was described in Section 1.9.6 and allows the user to obtain a hardcopy of the statistical results of the program giving details of the factors, standard deviations and correlation coefficients as appropriate. This routine is called by issuing a **GOTO 6500** command. The last line can either return the user to the program or exit, as shown, to the end of the program.

Data analysis printout routine—BASIC routine

```
6500 LOCATE 24, 10: PRINT " ";
6510 LOCATE 24, 10: PRINT "Switch Printer Online - Continue (Y/N) ";
6520 P$ = INPUT$(1)
6530 IF INSTR("YNyn", P$) = 0 THEN 6500
6540 IF P$ = "N" OR P$ = "n" THEN 9530
6550 LPRINT PC$
6560 LPRINT "DATA File : "; DFILEN$: LPRINT TITLE$
6570 LPRINT "****************************************"
6580 LPRINT "Factor A = "; A
6590 LPRINT "Factor B = "; B
6600 LPRINT "Mean X = "; XBAR
6610 LPRINT "Mean Y = "; YBAR
6620 LPRINT "Standard Deviation X = "; STDX
6630 LPRINT "Standard Deviation Y = "; STDY
6640 LPRINT "Coefficient of Correlation = "; RXY
6650 LPRINT "Determination Coefficient = "; RXY ^ 2
6700 LPRINT "===================================="
6710 LPRINT TAB(5); "X-Data" ; TAB(15) ; "Y-Data" ; TAB(25) ; "Y-
   Calc";
6715 LPRINT TAB(33); "Residual"
6720 LPRINT "===================================="
6730 FOR I = 1 TO NROW
6740 REM
6750 Y = FACTA + FACTB * Z(I, 1)
6760 REM
6770 LPRINT USING "######.###"; Z(I, 1); Z(I, 2); Y; Z(I, 2) - Y
6780 NEXT I
6790 LPRINT "****************************************"
6800 LPRINT
```

```
6810 GOTO 9530
```

APPENDIX A.8:
DATA CHECK FOR MAXIMUM AND MINIMUM VALUES COMPUTER ROUTINE AT LINE 7000

This routine was described in Section 1.9.7 and reads through the data file to find the maximum and minimum values of *X* (**XMAX** and **XMIN**) and *Y* (**YMAX** and **YMIN**). This routine is called by issuing the command **GOSUB 7000** and hence the **RETURN** statement on the last line. During the running of this routine a note is displayed on the screen "Please wait…checking DATA", to advise the user accordingly.

Data check for maximum and minimum values—BASIC routine

```
7000 LOCATE 6, 10: PRINT "Please wait ..... checking DATA"
7010 XMIN = Z(1, 1): XMAX = Z(NROW, 1)
7020 YMIN = 25000: YMAX = -25000
7030 FOR I2 = 1 TO NROW
7040 FOR J2 = 2 TO NCOL
7050 IF Z(I2, J2) < YMIN THEN YMIN = Z(I2, J2)
7060 IF Z(I2, J2) > YMAX THEN YMAX = Z(I2, J2)
7070 NEXT J2
7080 NEXT I2
7090 RETURN
```

APPENDIX A.9:
GRAPH AXES INTERVALS COMPUTER ROUTINE AT LINE 7100

This routine was described in Section 1.9.8 and computes the most suitable scaling intervals on the graph axes. Initially in each program, before calling this routine, the intervals to give ten divisions are calculated **(XTC** and **YTC)** and these are stored sequentially in the variable **TCC** which is passed to this routine. The value of **TCC** is examined and the nearest equitable division chosen, i.e. 1, 2.5, 5, 10, etc. This rounded up value of **TCC** is returned to the program and passed to **XTC** and **YTC** before graph scaling. This routine is called by issuing the command **GOSUB 7100** and hence the **RETURN** on the last line. Some 19 different divisions are given in this routine, but these can be changed as required by the user.

Graph axes intervals—BASIC routine

```
7100 IF TCC <= .01 THEN TCC = .01: GOTO 7290
7110 IF TCC <= .025 THEN TCC = .025: GOTO 7290
7120 IF TCC <= .05 THEN TCC = .05: GOTO 7290
7130 IF TCC <= .1 THEN TCC = .1: GOTO 7290
7140 IF TCC <= .25 THEN TCC = .25: GOTO 7290
7150 IF TCC <= .5 THEN TCC = .5: GOTO 7290
7160 IF TCC <= 1 THEN TCC = 1: GOTO 7290
7170 IF TCC <= 2.5 THEN TCC = 2.5: GOTO 7290
7180 IF TCC <= 5 THEN TCC = 5: GOTO 7290
7190 IF TCC <= 10 THEN TCC = 10: GOTO 7290
7195 IF TCC <= 15 THEN TCC = 15: GOTO 7290
```

```
7200 IF TCC <= 20 THEN TCC = 20: GOTO 7290
7210 IF TCC <= 25 THEN TCC = 25: GOTO 7290
7220 IF TCC <= 30 THEN TCC = 30: GOTO 7290
7230 IF TCC <= 40 THEN TCC = 40: GOTO 7290
7240 IF TCC <= 50 THEN TCC = 50: GOTO 7290
7250 IF TCC <= 100 THEN TCC = 100: GOTO 7290
7260 IF TCC <= 250 THEN TCC = 250: GOTO 7290
7270 IF TCC <= 500 THEN TCC = 500: GOTO 7290
7280 TCC = 1000
7290 RETURN
```

APPENDIX A.10:
SET UP GRAPHICS SCREEN TEMPLATE COMPUTER ROUTINE AT LINE 7300

This routine was described in Section 1.9.9 and provides a standard template for graph plotting. The intersection of the *X*-axis and *Y*-axis is given before calling this routine **(X0=70:Y0=50)** and the graph axes drawn within a border or boundary line. This routine is called by all the graph plotting programs by issuing the command **GOSUB 7300**. If only a border is required for pie charts or 3-D surface models, then **GRAPH=0** and the program returns from Line 7350. Where only an axis is required with no divisions, then **GRAPH=1** and the program returns from Line 7380. If, however, both *X*-axis and *Y*-axis divisions are to be plotted, then **GRAPH=2,** and the graph axes are labelled from **XMIN** to **XMAX** at the intervals chosen in Appendix A.9. Note that text labels are not used, but another routine at Line 55000 (Appendix A.23) called to draw the numeric values.

Set up graphics screen template—BASIC routine

```
7300 IF SCR% <> 2 THEN 7320
7310 COLOR 15, 0
7320 CLS : GOSUB 9000
7330 IF SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
7340 LINE (10, 20) - (620, 180), RGB%, B
7350 IF GRAPH = 0 THEN RETURN
7360 IF SCR% = 2 THEN RGB% = 15 ELSE RGB% = 3
7370 LINE (GX0, PY) - (GX0, GY0), RGB%: LINE - (PX, GY0), RGB%
7380 IF GRAPH = 1 THEN RETURN
7390 FOR LXD = 0 TO 10
7400 AVGX = GX0 + LXD * (PX - GX) / 10: AVGY = GY0 - 11
7405 LDAT = INT( (XMIN + LXD * XTC) * 1000 + .5) / 1000
7410 PSET (AVGX, GY0) : LINE -(AVGX, GY0 - 3), RGB%
7420 AVGX = AVGX - 6: GOSUB 55000
7430 NEXT LXD
7440 FOR LYD = 0 TO 10
7450 AVGY = GY0 + LYD * (PY - GY) / 10: AVGX = GX0 - 32
7455 LDAT = INT( (YMIN + LYD * YTC) * 1000 + .5) / 1000
7460 PSET (GX0, AVGY): LINE -(GX0 - 7, AVGY), RGB%
7470 AVGY = AVGY - 4: GOSUB 55000
7480 NEXT LYD
7490 RETURN
```

APPENDIX A.11:
DISPLAY GRAPH TEXT LABELS COMPUTER ROUTINE AT LINE 7500

This routine was described in Section 1.9.10 and provides the graph with a title and the axes with labels. Both the title **(TITLE$)** and the *X*-axis label **(LXAXI$)** are placed centrally, with the *Y*-axis label **(LYAXI$)** either placed vertically or horizontally as chosen by the user. Note, if a background grid is displayed, then only a *Y*-axis vertical label can be used.

Display graph text labels—BASIC routine

```
7500 LNT = LEN(TITLE$) : LNC = 40 – LNT / 2
7510 LOCATE 2, LNC: PRINT TITLE$
7520 LNT = LEN(LXAXI$): LNC = 40 – LNT / 2
7530 LOCATE 21, LNC: PRINT LXAXI$
7535 IF YLAB$ = "V" THEN 7550
7540 LOCATE 4, 11: PRINT LYAXI$
7545 RETURN
7550 YDIST = LEN(LYAXI$)
7560 FOR I = 1 TO YDIST
7570 LOCATE 6 + I, 3: PRINT MID$(LYAXI$, I, 1)
7580 NEXT I
7590 RETURN
```

APPENDIX A.12:
DISPLAY GRAPH AXIS GRID COMPUTER ROUTINE AT LINE 7600

This routine was described in Section 1.9.11. The background grid is drawn at each of the axes divisions selected in Appendix A.9. The user can choose in the program to have a horizontal grid **(HGRID=1)** with or without a vertical grid. If no vertical grid is required **(VGRID=0),** then the routine returns from Line 7660. If a vertical grid is chosen **(VGRID=1)** then vertical lines are drawn as well. The default style and colour for the grid are dotted lines **(ST=&HCCCC)** in white, but this can be changed by altering the colour **(CL%)** and the style **(ST)** accordingly. This routine is called by issuing the command **GOSUB 7600** after the user has had the option to select a background grid.

Display graph axis grid—BASIC routine

```
7600 IF SCR% = 2 THEN CL% = 15 ELSE CL% = 0
7610 ST = &HCCCC
7620 FOR LYD = 1 TO 10
7630 AVGY = Y0 + LYD * 12
7640 PSET (X0, AVGY): LINE – (X0 + XM, AVGY), CL%, , ST
7650 NEXT LYD
7660 IF VGRID = 0 THEN RETURN
7670 FOR LXD = 1 TO 10
7680 AVGX = X0 + LXD * 50
7690 PSET (AVGX, Y0): LINE – (AVGX, Y0 + YM), CL%, , ST
7700 NEXT LXD
7710 RETURN
```

APPENDIX A.13:
SET UP COLOUR SHADED PANEL COMPUTER ROUTINE AT LINE 7800

This routine was described in Section 1.9.12. It is used in conjunction with the programs using colour shading, e.g., contouring, 3-D profiles and shaded mesh, to indicate the average level of the spectral mapped cell by colour. A panel is drawn on the right-hand side of the screen, together with the associated contour levels from the highest **(HC)** to lowest **(LC)**. This routine is called by issuing the command **GOSUB 7800** prior to plotting the contours or surface models.

Set up colour shaded panel—BASIC routine

```
7800 LINE (260 * SR%, GY) – (280 * SR%, GY – 45 + 167), RGB%, B
7810 FOR IB = 0 TO NOCL + 1
7815 YT1 = GY – 45 + 167 – IB * 8: YT2 = GY – 45 + 167 – IB * 8
7820 LINE (260 * SR%, YT1) – (281 * SR%, YT2), RGB%
7830 NEXT IB
7840 FOR JC = 0 TO NOCL
7850 PAINT (270 * SR%, GY – 45 + 163 – JC * 8), CL% (JC), RGB%
7860 NEXT JC
7870 FOR KC = 0 TO NOCL
7880 CCL = INT( (HC – KC * TCC) * 100) / 100
7890 LOCATE (5 + 1 * KC), (36 * SR%) : PRINT CCL
7900 NEXT KC
7910 RETURN
```

APPENDIX A.14:
SET UP MONOCHROME SHADING COMPUTER ROUTINE AT LINE 8600

This routine was described in Section 1.9.13 and provides different types of shading for the screen display of charts and graphs. Five different pixel patterns are supplied for five data sets and the patterns stored in a string array **TILE$(I)**. These patterns can be altered by the user if required. This routine is called by issuing the command **GOSUB 8600** after the necessary array has been dimensioned earlier in the program.

Set up monochrome shading—BASIC routine

```
8599 REM ***** SET UP TILES FOR B & W SCREEN *****
8600 TILE$(1) = CHR$(&HFF) + CHR$(&HFF)
8610 TILE$(2) = CHR$(&H3C) + CHR$(&H3C) + CHR$(&H3C) + CHR$(&H3C)
8620 TILE$(2) = TILE$(2) + CHR$(&HC3) + CHR$(&HC3) + CHR$(&HC3) +
   CHR$(&HC3)
8630 TILE$(3) = CHR$(&HFF) + CHR$(&H0)
8640 TILE$(4) = CHR$(&HCC) + CHR$(&HCC) + CHR$(&H33) + CHR$(&H33)
8650 TILE$(5) = CHR$(&H55) + CHR$(&HAA) + CHR$(&H55) + CHR$(&HAA)
8660 RETURN
```

APPENDIX A.15:
SCREEN HEADER COMPUTER ROUTINE AT LINE 9000

This routine was described in Section 1.9.14 and serves to annotate each computer display. The type of graph, e.g. “Line Graph” should be stored in the string variable **PC$** before calling this routine using the command **GOSUB 9000**. The date will also be displayed **(DATE$)** together with the current data file **(DFILEN$)** at the top of the screen for reference purposes.

Screen header—BASIC routine

```
9000 T$ = "Presentation Graphics"
9010 LOCATE 1, 2 * SR%: PRINT T$
9020 LOCATE 1, 31 * SR%: PRINT DATE$
9030 LOCATE 2, 2 * SR%: PRINT PC$
9040 LOCATE 2, 31 * SR%: PRINT TIME$
9050 LOCATE 3, 1: FOR K = 1 TO 40 * SR%: PRINT "="; : NEXT K
9060 PRINT
9070 IF TITLE = 1 THEN 9090
9080 GOTO 9100
9090 RETURN
9095 REM ***** PC-Presentation Graphics Title Page *****
9100 FOR I = 6 TO 20
9110 LOCATE I, I - 4: PRINT "PC-Presentation Graphics";
9120 IF SR% = 1 THEN 9140
9130 LOCATE I, 36 + 1: PRINT "PC-Presentation Graphics"
9140 NEXT I
9150 LOCATE 23, 25: PRINT "Copyright (C) 1991 by P.H.Milne"
9160 TITLE = 1
9170 START = TIMER
9180 IF (TIMER - START) > 2 THEN 9190 ELSE 9180
9190 CLS : GOTO 9000
```

APPENDIX A.16:
PROGRAM END COMPUTER ROUTINE AT LINE 9500

This routine was described in Section 1.9.15 and gives the user an opportunity to run the program again with a different type of presentation or data file, or of returning to the master menu. Each of the graphics presentation programs use this routine which awaits the press of the space bar before giving the user the above option which is displayed centrally at the foot of the screen. This allows the user either to save the screen to disc or as a screen dump to a printer without including any unnecessary text.

Program end—BASIC routine

```
9500 Z$ = INPUT$(1)
9510 IF Z$ = "" THEN 9500 ELSE 9520
9520 GOSUB 9000
9530 LABEL$ = "Run Line Graph again (Y/N)"
9540 LNT = LEN(LABEL$): LNC = 40 - LNT/2
```

```
9550 LOCATE 24, LNC: PRINT LABEL$; : A$ = INPUT$(1)
9560 IF INSTR("YNyn", A$) = 0 THEN 9530
9570 IF A$ = "N" OR A$ = "n" THEN 9600
9580 RUN MDSK$ + "GRAPH-B1"
9600 RUN MDSK$ + "PG-MENU"
```

APPENDIX A.17:
ERROR ROUTINES COMPUTER ROUTINE AT LINE 10000

This routine was described in Section 1.9.16 and serves to trap disc and data errors. The error line numbers **(ERL)** may have to be altered, depending on which program is being used. These routines must be preceded by an **ON ERROR GOTO 10000** at the beginning of each program. If the program crashes due to a programming error on the user's part, the type of error **(ERR)** and the line number of its location **(ERL)** will be displayed on screen. The user will need to consult a BASIC manual to determine the type of error to correct the program line accordingly.

Error routines—BASIC routine

```
10000 IF ERR = 71 THEN 10100
10010 IF ERR = 53 AND ERL = 8500 THEN RESUME 500
10020 IF ERR = 53 THEN 10200
10030 IF ERL = 8000 THEN 10200
10040 IF ERL = 8220 THEN 10300
10050 IF ERL = 8300 THEN 10300
10080 PRINT : PRINT "Error No. = "; ERR; "on Line No. = "; ERL
10090 GOTO 10090
10100 BEEP
10110 LOCATE 22, 10: PRINT "Disc not Ready - Insert Disc in Drive"
10120 LOCATE 23, 10: INPUT "Press Return to Continue "; R$
10130 LOCATE 22, 10: PRINT " "
10140 LOCATE 23, 10: PRINT " "
10150 RESUME 0
10200 BEEP: CLS : GOSUB 9000
10210 LOCATE 10, 10: PRINT "File not Found"
10220 LOCATE 15, 10: PRINT "You Must Install Setup First"
10230 LOCATE 20, 10: INPUT "Press Return to Continue "; R$
10240 RUN MDSK$ + "PG-MENU"
10300 LOCATE 10, 5: PRINT "You Must Load DATA File First"
10310 GOTO 10230
```

APPENDIX A.18:
DATA SYMBOL DISPLAY COMPUTER ROUTINE AT LINE 40000

This routine was described in Section 1.9.17 and provides five different types of symbols for identifying line graphs of different data sets. A high line number keeps it free from any other programming routines. The computer listing includes **REM**ark lines for symbol identification. If the user wishes to alter the size of the symbols, this can be done by altering the values of **W** for width and **H** for height—note these values must be integer values as they are dealing with pixel positions.

Data symbol display—BASIC routine

```
39999 REM ***** Symbols for Graphs *****
40000 W = 3: H = 2
40010 ON J GOTO 40100, 40200, 40300, 40400, 40500
40099 REM ***** 1 - Square Box *****
40100 IF SCR% = 2 THEN CL%(1) = 4 ELSE CL%(1) = 1
40105 IF SCR% = 2 THEN CL%(6) = 12 ELSE CL%(6) = 1: GOTO 40170
40110 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H) , CL%(1), BF
40120 RETURN
40170 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), , B
40180 PAINT (AVGX, AVGY), TILE$(1)
40190 RETURN
40199 REM ***** 2 - Triangle *****
40200 IF SCR% = 2 THEN CL%(2) = 1 ELSE CL%(2) = 1
40205 IF SCR% = 2 THEN CL%(7) = 9 ELSE CL%(7) = 1: GOTO 40270
40210 IF DATBX – 1 THEN 40260
40220 LINE (AVGX, AVGY + H) – (AVGX – W, AVGY – H), CL%(2)
40230 LINE – (AVGX + W, AVGY – H) , CL%(2): LINE – (AVGX, AVGY + H),
   CL%(2)
40240 PAINT (AVGX, AVGY), CL% (2)
40250 RETURN
40260 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), CL%(2), BF:
   RETURN
40270 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), , B
40280 PAINT (AVGX, AVGY), TILE$(2)
40290 RETURN
40299 REM ***** 3 - Diamond *****
40300 IF SCR% = 2 THEN CL% (3) = 5 ELSE CL% (3) = 1
40305 IF SCR% = 2 THEN CL% (8) = 13 ELSE CL%(8) = 1: GOTO 40370
40310 IF DATBX = 1 THEN 40360
40320 LINE (AVGX, AVGY + H) – (AVGX – W, AVGY), CL% (3)
40330 LINE – (AVGX, AVGY – H) , CL%(3): LINE – (AVGX + W, AVGY), CL%
   (3)
40340 LINE – (AVGX, AVGY+H) , CL% (3)
40350 PAINT (AVGX, AVGY), CL%(3): RETURN
40360 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), CL%(3), BF:
   RETURN
40370 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), , B
40380 PAINT (AVGX, AVGY), TILE$(3)
40390 RETURN
40399 REM ***** 4 - Cross 'x' sign *****
40400 IF SCR% = 2 THEN CL% (4) = 3 ELSE CL% (4) = 1
40405 IF SCR% = 2 THEN CL% (9) = 11 ELSE CL% (9) = 1: GOTO 40470
40410 IF DATBX = 1 THEN 40450
40420 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), CL%(4)
40430 LINE (AVGX – W, AVGY + H) – (AVGX + W, AVGY - H), CL%(4)
40440 RETURN
40450 LINE (AVGX – W, AVGY – H) – (AVGX + W, AVGY + H), CL%(4), BF
```

```
40460 RETURN
40470 LINE (AVGX - W, AVGY - H) - (AVGX + W, AVGY + H), , B
40480 PAINT (AVGX, AVGY), TILE$(4)
40490 RETURN
40499 REM ***** 5 - Plus '+' sign *****
40500 IF SCR% = 2 THEN CL%(5) = 2 ELSE CL%(5) = 1
40505 IF SCR% = 2 THEN CL%(10) = 10 ELSE CL%(10) = 1: GOTO 40570
40510 IF DATBX = 1 THEN 40550
40520 LINE (AVGX, AVGY - H) - (AVGX, AVGY + H) , CL%(5)
40530 LINE (AVGX - W, AVGY) - (AVGX + W, AVGY), CL%(5)
40540 RETURN
40550 LINE (AVGX - W, AVGY - H) - (AVGX + W, AVGY + H), CL%(5), BF
40560 RETURN
40570 LINE (AVGX - W, AVGY - H) - (AVGX + W, AVGY + H), , B
40580 PAINT (AVGX, AVGY), TILE$(5)
40590 RETURN
```

APPENDIX A.19:

PLOTTER ROUTINES COMPUTER ROUTINE AT LINE 50000

This routine was described in Section 1.9.18 and allows use of an HPGL plotter.

Plotter routines—BASIC routine

```
50000 CLS : GOSUB 9000
50010 LOCATE 8, 5: PRINT "Select Plotter :-"
50020 LOCATE 10, 5: PRINT "<1> Hewlett-Packard A4"
50030 LOCATE 12, 5: PRINT "<2> Hewlett-Packard A3"
50040 LOCATE 14, 5: PRINT "<3> Hewlett-Packard A1"
50050 LOCATE 16, 5: PRINT "<4> Return to Menu: PLOT$ = INPUT$(1)
50060 IF INSTR("1234", PLOT$) = 0 THEN 50050
50065 PLOT% = VAL(PLOT$)
50070 ON PLOT% GOTO 50075, 50080, 50090, 9520
50075 IPXMAX = 10800: IPYMAX = 6800: IPXMIN = 0: IPYMIN = 0
50076 XMM = .27: YMM = .17: TSEC = 1: GOTO 50100
50080 IPXMAX = 15200: IPYMAX = 10800: IPXMIN = 0: IPYMIN = 0
50081 XMM = .38: YMM = .27: TSEC = 1: GOTO 50100
50090 IPXMAX = 15200: IPYMAX = 10800: IPXMIN = -15200: IPYMIN =
   -10800
50091 XMM = .74: YMM = .48: TSEC = 1
50100 LOCATE 19, 5:
50101 PRINT "Select Interface <1> LPT1:, <2> LPT2:, or <3> COM1:"
50105 LOCATE 20, 5:
50106 PRINT "or <4> HP Plot File Output";
50107 PLF$ = INPUT$(1)
50110 IF INSTR("1234", PLF$) = 0 THEN 50105
50115 PLF% = VAL(PLF$)
50120 ON PLF% GOTO 50130, 50140, 50150, 50151
50130 OPEN "LPT1:" FOR OUTPUT AS #3: GOTO 50160
```

```
50140 OPEN "LPT2:" FOR OUTPUT AS #3: GOTO 50160
50150 OPEN "COM1:1200,E,7,1,CS2000,DS15000" FOR OUTPUT AS #3: GOTO
   50160
50151 TSEC = 0:
50152 LOCATE 22, 5: PRINT "Enter Plot File Name "; : INPUT HPFILE$
50153 IF HPFILE$ = "" THEN 50152
50154 OPEN "O", #3, DDSK$ + HPFILE$ + ".HPF"
50155 RETURN
50160 LOCATE 22, 5: PRINT "Plotter Ready to Plot (Y/N) ";
50165 QQ$ = INPUT$(1)
50170 IF INSTR("Y", QQ$) = 0 THEN 50160
50180 RETURN
```

APPENDIX A.20:
PLOTTER TITLE COMPUTER ROUTINE AT LINE 50500

This routine was described in Section 1.9.19 and allows use of an HPGL plotter. The pen colour can be changed by altering the **SP** number.

Plotter title—BASIC routine

```
50500 PRINT #3, "PU;SP1;SI0.25,0.4;": GOSUB 60000
50510 PRINT #3, "PA10,194;LB"; T$; CHR$(3): GOSUB 60000
50520 PRINT #3, "PU;PA620,194;CP"; -LEN(DATE$); "0;": GOSUB 60000
50530 PRINT #3, "LB"; DATE$; CHR$(3): GOSUB 60000
50540 PRINT #3, "PA10,184;LB"; PC$; CHR$(3): GOSUB 60000
50550 PRINT #3, "PU;PA620,184;CP"; -LEN(DFILEN$); "0;": GOSUB 60000
50560 PRINT #3, "LB"; DFILEN$; CHR$(3): GOSUB 60000
50570 RETURN
```

APPENDIX A.21:
PLOTTER LINE SPECIFICATION COMPUTER ROUTINE AT LINE 51000

This routine was described in Section 1.9.20 and allows use of an HPGL plotter. Only four pen numbers (1–4) have been given in the **SP** command for use with standard 4-pen plotters, but users with 8-pen plotters can change these numbers as required. The line type can also be changed by altering the **LT** specifications.

Plotter line specification—BASIC routine

```
51000 ON J GOTO 51001, 51002, 51003, 51004, 51005
51001 PRINT #3, "PU;SP2;" : GOSUB 60000: GOTO 51010
51002 PRINT #3, "PU;SP4;" : GOSUB 60000: GOTO 51010
51003 PRINT #3, "PU;SP1;" : GOSUB 60000: GOTO 51010
51004 PRINT #3, "PU;SP4;" : GOSUB 60000: GOTO 51010
51005 PRINT #3, "PU;SP3;" : GOSUB 60000: GOTO 51010
51010 REM - Select Line Type
51030 ON J GOTO 51031, 51032, 51033, 51034, 51035
51031 PRINT #3, "LT;": GOTO 51040
```

```
51032 PRINT #3, "LT3, 2;": GOTO 51040
51033 PRINT #3, "LT3, 3;": GOTO 51040
51034 PRINT #3, "LT3, 4;": GOTO 51040
51035 PRINT #3, "LT4, 2;": GOTO 51040
51040 PRINT #3, "PU;PA"; X1; ","; Y1; ",": GOSUB 60000
51050 PRINT #3, "PD;PA"; X2; Y2; ",": GOSUB 60000
51060 RETURN
```

APPENDIX A.22:
PLOTTER SHADING SPECIFICATION COMPUTER ROUTINE AT LINE 52000

This routine was described in Section 1.9.21 and allows use of an HPGL plotter. As in Appendix A.21 only four pen numbers have been used. The user can also change the **FT** setting for shading as required after consulting their plotter manual for **FT** commands.

Plotter shading specification—BASIC routine

```
52000 ON (K - 1) GOTO 52001, 52002, 52003, 52004, 52005
52001 PRINT #3, "PU;SP2;" : GOSUB 60000: GOTO 52010
52002 PRINT #3, "PU;SP4;" : GOSUB 60000: GOTO 52010
52003 PRINT #3, "PU;SP1;" : GOSUB 60000: GOTO 52010
52004 PRINT #3, "PU;SP4;" : GOSUB 60000: GOTO 52010
52005 PRINT #3, "PU;SP3;" : GOSUB 60000: GOTO 52010
52010 PRINT #3, "PU;PA"; X1; "," ; Y1; ";" : GOSUB 60000
52020 PRINT #3, "EA"; X2; ","; Y2; ";": GOSUB 60000
52030 ON (K-1) GOTO 52031, 52032, 52033, 52034, 52035
52031 PRINT #3, "FT4, 3, 0;": GOTO 52040
52032 PRINT #3, "FT4, 3, 45;": GOTO 52040
52033 PRINT #3, "FT4, 3, 0;": GOTO 52040
52034 PRINT #3, "FT3, 3, -45;": GOTO 52040
52035 PRINT #3, "FT3,5, 45;": GOTO 52040
52040 PRINT #3, "RA"; X2; ","; Y2; ";": GOSUB 60000
52050 RETURN
```

APPENDIX A.23:
GRAPHICS NUMBERS COMPUTER ROUTINE AT LINE 55000

This routine was described in Section 1.9.22 and draws the numbers 0 to 9 on screen for labelling the axes and identifying contour levels. The size of the numbers can be altered by changing the values of **W** and **H** as mentioned in Appendix A.18.

Graphics numbers—BASIC routine

```
54999 REM ***** Graphics Numbers *****
55000 K$ = STR$(LDAT)
55110 IF LDAT < 0 THEN NEG = 1 ELSE NEG = 0
55150 LN = LEN(K$)
55160 N1$ = MID$(K$, 2, 1)
```

```
55170 N2$ = MID$(K$, 3, 1)
55180 N3$ = MID$(K$, 4, 1)
55190 N4$ = MID$(K$, 5, 1)
55200 MX = 1: TNL = 1: NX = 0: W = 3 * MX: H = 2 * MX
55205 IF NEG = 0 THEN 55220
55210 XP = AVGX + (2 + NX - 6) * MX: YP = AVGY + 2 * MX
55215 LINE (XP, YP + H) - (XP + W, YP + H)
55220 IF N1$ = "." THEN PSET (AVGX + (3 + NX) * MX, AVGY + 2 * MX)
55221 GOTO 55240
55225 N1 = VAL(N1$)
55230 ON (N1 + 1) GOSUB 56000, 56100, 56200, 56300, 56400, 56500,
   56600, 56700,
56800, 56900
55240 IF TNL = (LN - 1) THEN RETURN
55250 TNL = TNL + 1: NX = 6
55255 IF N2$ = "." THEN PSET (AVGX + (3 + NX) * MX, AVGY + 2 * MX)
55256 GOTO 55280
55260 N2 = VAL(N2$)
55270 ON (N2 + 1) GOSUB 56000, 56100, 56200, 56300, 56400, 56500,
   56600, 56700,
56800, 56900
55280 IF TNL = (LN - 1) THEN RETURN
55290 TNL = TNL + 1: NX = 12
55295 IF N3$ = "." THEN PSET (AVGX + (3 + NX) * MX, AVGY + 2 * MX)
55296 GOTO 55320
55300 N3 = VAL(N3$)
55310 ON (N3 + 1) GOSUB 56000, 56100, 56200, 56300, 56400, 56500,
   56600, 56700,
56800, 56900
55320 IF TNL = (LN - 1) THEN RETURN
55330 TNL = TNL + 1: NX = 18
55335 IF N4$ = "." THEN PSET (AVGX + (3 + NX) * MX, AVGY + 2 * MX)
55336 GOTO 55360
55340 N4 = VAL(N4$)
55350 ON (N4 + 1) GOSUB 56000, 56100, 56200, 56300, 56400, 56500,
   56600, 56700.
56800, 56900
55360 RETURN
56000 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56010 LINE (XP, YP) - (XP + W, YP), RGB%
56020 LINE (XP + W, YP) - (XP + W, YP + 2 * H) , RGB%
56030 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H) , RGB%
56040 LINE (XP, YP+2 * H) - (XP, YP) , RGB%
56060 RETURN
56100 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56110 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56160 RETURN
56200 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56210 LINE (XP + W, YP) - (XP, YP), RGB%
```

```
56220 LINE (XP, YP) - (XP, YP + H), RGB%
56230 LINE (XP, YP + H) - (XP + W, YP + H), RGB%
56240 LINE (XP + W, YP + H) - (XP + W, YP + 2 * H), RGB%
56250 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H), RGB%
56260 RETURN
56300 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56310 LINE (XP, YP) - (XP + W, YP), RGB%
56320 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56330 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H) , RGB%
56340 LINE (XP, YP + H) - (XP + W, YP + H), RGB%
56360 RETURN
56400 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56410 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56420 LINE (XP + W, YP + H) - (XP, YP + H), RGB%
56430 LINE (XP + W, YP + H) - (XP, YP + H), RGB%
56440 LINE (XP, YP + H) - (XP, YP + 2 * H), RGB%
56460 RETURN
56500 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56510 LINE (XP, YP) - (XP + W, YP), RGB%
56520 LINE (XP + W, YP) - (XP + W, YP + H), RGB%
56530 LINE (XP + W, YP + H) - (XP, YP + H), RGB%
56540 LINE (XP, YP + H) - (XP, YP + 2 * H) , RGB%
56550 LINE (XP, YP + 2 * H) - (XP + W, YP + 2 * H)
56560 RETURN
56600 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56610 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H), RGB%
56620 LINE (XP, YP + 2 * H) - (XP, YP), RGB%
56630 LINE (XP, YP) - (XP + W, YP), RGB%
56640 LINE (XP + W, YP) - (XP + W, YP + H), RGB%
56650 LINE (XP + W, YP + H) - (XP, YP + H), RGB%
56660 RETURN
56700 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56710 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56720 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H), RGB%
56760 RETURN
56800 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56810 LINE (XP, YP) - (XP + W, YP), RGB%
56820 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56830 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H), RGB%
56840 LINE (XP, YP + 2 * H) - (XP, YP), RGB%
56850 LINE (XP, YP + H) - (XP + W, YP + H), RGB%
56860 RETURN
56900 XP = AVGX + (2 + NX) * MX: YP = AVGY + 2 * MX
56910 LINE (XP, YP) - (XP + W, YP), RGB%
56920 LINE (XP + W, YP) - (XP + W, YP + 2 * H), RGB%
56930 LINE (XP + W, YP + 2 * H) - (XP, YP + 2 * H), RGB%
56940 LINE (XP, YP + 2 * H) - (XP, YP + H), RGB%
56950 LINE (XP, YP + H) - (XP + W, YP + H), RGB%
56960 RETURN
```

APPENDIX A.24:
DELAY ROUTINE FOR PLOTTERS COMPUTER ROUTINE AT LINE 60000

This routine was described in Section 1.9.23 and allows for microcomputers which send data too quickly for the plotter to respond. The value of **TSEC=1** is set earlier in the program before calling this routine, and will depend on the particular combination of microcomputer and plotter used. If the plotter is very slow with the user's set-up, then **TSEC** can be reduced. Alternatively, if lines are lost, **TSEC** should be increased.

Delay routine for plotters—BASIC routine

```
60000 REM DELAY TIME ROUTINE FOR HPGL PLOTTERS
60010 REM FOR TSEC Seconds
60020 GOSUB 60100
60030 VALUET = VALUE + TSEC
60040 GOSUB 60100
60050 IF VALUE < VALUET THEN 60040
60060 RETURN
60100 VALUE$ = TIME$
60110 VALUER = VAL(LEFT$(VALUE$, 2))
60120 VALUEM = VAL(MID$(VALUE$, 4, 2))
60130 VALUES = VAL(RIGHT$(VALUE$, 2))
60140 VALUE = VALUEH * 3600 + VALUEM * 60 + VALUES
60150 RETURN
```

APPENDIX A.25:
KEYBOARD DATA ENTRY COMPUTER PROGRAM "KEYBDATA"

This program is described in Section 1.10.1 and is used for the keyboard entry of a **'.DAT'** data file.

Keyboard data entry

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, the numeric and string variables which require to be initialized and the error trap location. The array **Z** is dimensioned for 100 rows of data with 5 data sets. If you wish to enter more data, this array should be re-dimensioned accordingly. Before anything is displayed on screen, the computer set-up is checked in the routine at Line 8000.

(b) Data file entry Line numbers 100–540

Data are entered in a spreadsheet format where **ND** is the number of data sets and **NROW** the number of rows of data. The *X*-axis data are entered in the first column. After all the data has been entered, the data are saved to disc with a **'.DAT'** extension, and an **'IDFILE.PGD'** file opened to contain the name of the file and its extension.

(c) Text file entry Line numbers 1000–1090

The user now has an opportunity to enter a graph title **TITLE$,** axes labels, **LXAXI$** and **LYAXI$,** and legends for each of the data series **DAT$(I)**. This information is then stored on disc using the **GOSUB 8400** routine described in Appendix A.4.

(d) Data file printout Line numbers 1100–1150

The user is now given an opportunity to obtain a data and text file printout using the **GOTO 6000** routine described in Appendix A.6.

This program also requires the following routines:

(e) Appendix A.6	Line numbers 6000–6320
(f) Appendix A.2	Line numbers 8000–8180
(g) Appendix A.4	Line numbers 8400–8490
(h) Appendix A.15	Line numbers 9000–9190
(i) Appendix A.16	Line number 9600 only
(j) Appendix A.17	Line numbers 10000–10240

KEYBDATA—BASIC program

```
10 REM <KEYBDATA> Presentation Graphics Keyboard DATA Entry
14 REM (C) Copyright P.H.Milne 1990 : Appendix A.25
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990
30 CLEAR : DIM Z(100, 6): GROW = 1: SR% = 2
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 CLS : PC$ = "Keyboard Input": GOSUB 9000
110 LOCATE 6, 10: INPUT "Enter DATA File Name - "; DFILEN$
120 LOCATE 7, 10: INPUT "Enter Number of DATA Sets "; ND
130 LOCATE 8, 10: INPUT "Enter Number of DATA Points "; NROW
140 NCOL = ND + 1
150 FOR D = 1 TO NCOL - 1
160 LOCATE 10, 10: PRINT "X-AXIS"
165 LOCATE 10, 10 + 10 * D: PRINT "DATA"; D
170 NEXT D
180 FOR I = 1 TO NROW
190 FOR J = 1 TO NCOL
200 LOCATE 23, 10: PRINT "Enter DATA"
210 LOCATE 23, 20: INPUT Z(I, J)
220 LOCATE 10 + GROW, 10 + (J – 1) * 10: PRINT Z(I, J)
230 NEXT J: GROW = GROW + 1
240 IF I/11 = INT(I/11) THEN 250 ELSE 290
250 FOR CROW = 1 TO 11
260 LOCATE 10 + CROW, 10
270 FOR K = 1 TO 70: PRINT " "; : NEXT K
```

```
280 NEXT CROW: GROW = 1
290 NEXT I
300 CLS : GOSUB 9000
310 LOCATE 6, 10: PRINT "Please wait ..... saving DATA"
320 OPEN "O", #1, DDSK$ + DFILEN$ + ".DAT"
330 PRINT #1, NROW, NCOL
340 FOR I = 1 TO NROW
350 FOR J = 1 TO NCOL - 1
360 PRINT #1, USING "######.###"; Z(I, J);
370 NEXT J
380 PRINT #1, USING "######.###"; Z(I, NCOL)
390 NEXT I
500 CLOSE
510 OPEN "O", #2, DDSK$ + "IDFILE.PGD"
520 PRINT #2, DFILEN$
530 PRINT #2, ".DAT"
540 CLOSE #2
1000 CLS : GOSUB 9000
1010 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
1020 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 40) "; LXAXI$
1030 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
1040 LOCATE 12, 10:
1041 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
1042 IF INSTR("HVhv", YLAB$) = 0 THEN 1040
1043 IF YLAB$ = "h" THEN YLAB$ = "H"
1044 IF YLAB$ = "v" THEN YLAB$ = "V"
1050 FOR I = 1 TO NCOL - 1
1060 LOCATE 13 + I, 10:
1061 PRINT "Enter Legend DATA "; I; " (Max 9) ";
1070 INPUT DAT$(I)
1080 NEXT I
1090 GOSUB 8400: REM Store Titles on Disc
1100 REM *** DATA Printout ***
1110 CLS : GOSUB 9000
1120 LOCATE 6, 10: PRINT "Do you wish to print DATA (Y/N) ";
1130 A$ = INPUT$(1)
1140 IF INSTR("YNyn", A$) = 0 THEN 1130
1150 IF A$ = "N" OR A$ = "n" THEN 9600
```

"KEYBDATA"—computer printout for data file 'PHMTEST2'

```
Presentation Graphics
Disc Data Input
DATA File - PHMTEST2
**********************
```

X-AXIS	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
0.000	5.000	10.000	15.000	20.000	25.000

X-AXIS	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
10.000	10.000	20.000	30.000	40.000	50.000
20.000	13.000	25.000	38.000	49.000	55.000
30.000	20.000	33.000	42.000	58.000	65.000
40.000	25.000	40.000	50.000	65.000	75.000
50.000	20.000	35.000	46.000	54.000	65.000
60.000	18.000	38.000	41.000	50.000	60.000
70.000	23.000	42.000	47.000	57.000	68.000
80.000	28.000	50.000	58.000	61.000	75.000
90.000	35.000	55.000	65.000	72.000	85.000
100.000	40.000	60.000	72.000	78.000	90.000

Graph Title - PC PRESENTATION GRAPHICS

```
X-AXIS Title - Days from Start of Business
Y-AXIS Title - No of Items
```

DATA	Legend
1	DATA 1
2	DATA 2
3	DATA 3
4	DATA 4
5	DATA 5

APPENDIX A.26:
DISC DATA RECALL COMPUTER PROGRAM “DISCDATA”

This program is described in Section 1.10.2 and is used either to reload a data file previously saved with a **‘.DAT’** extension as discussed in Appendix A.25, or to load an ASCII data file.

Disc data recall

(a) Initialization and control Line numbers 10–90

This segment of code is similar to section (a) of Appendix A.25.

(b) Data file recall Line numbers 100–950

Line numbers 120 to 440 of this routine are identical to that of section (b) of Appendix A.25, and can be reused. This routine opens at Line 100 with a request for the data filename and then jumps from Line 115 to 600 to recall the required data file. A check is first made on the disc drive and directory for the location of the data file. The data file extension is then requested. The program then attempts to open that file in Line 720. If this is unsuccessful and the file cannot be found, an error trap in Line 10010 resumes the program at Line 900 to allow the user to correct the file name or return to the initial menu.

If the file has been opened successfully, the extension is then checked, and if it is a contour **'.CON'** or surface **'.EQD'** data file it will already be in the correct format, etc., so the program returns to Line 400. If the file has a **'.DAT'** or other extension, it is then opened and the data read, with an opportunity to view the data on screen before saving to disc with a **'.DAT'** extension. After saving the data file, the routine at Line 400 then opens an identity file **'IDFILE.PGD'** for future reference. The **GOSUB 8500** routine at Line 550 then checks to see if a **'.TXT'** file exists with a graph title and axes labels. If there is no such file, an error trap at Line 10020 then resumes the program at Line 1000 to enter the data accordingly. If a **'.TXT'** file already exists, the program continues at Line 1100.

(c) Text file entry — Line numbers 1000–1090

This segment of code is identical to section (c) of Appendix A.25 and can be reused.

(d) Data file printout — Line numbers 1100–1150

This segment of code is similar to section (d) of Appendix A.25, the only difference being the **'.DAT'** extension check in Line 1100. Printouts can only be obtained for files with a **'.DAT'** extension, since contour and surface model files have too many rows and columns for efficient display. If the user wishes to view the contents of a **'.CON'** or **'.EQD'** file, the **TYPE** command should be used from MS-DOS, e.g. **TYPE DFILEN$.EXT$.** A printout can also be obtained by adding>**PRN** to the above **TYPE** statement.

This program also requires the following routines.

(e) Appendix A.6	Line numbers 6000–6320
(f) Appendix A.2	Line numbers 8000–8180
(g) Appendix A.4	Line numbers 8400–8490
(h) Appendix A.5	Line numbers 8500–8590
(i) Appendix A.15	Line numbers 9000–9190
(j) Appendix A.16	Line numbers 9600 only
(k) Appendix A.17	Line numbers 10000–10240

Note the addition of two error check Lines 10010 and 10020 as mentioned in section (b) above.

"DISCDATA"—BASIC program

```
10 REM <DISCDATA> Presentation Graphics DISC DATA Entry
14 REM (C) Copyright P.H.Milne 1990 : Appendix A.26
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990
30 CLEAR : DIM Z(100, 6): GROW = 1: SR% = 2
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 CLS : PC$ = "Disc Data Input": GOSUB 9000
110 LOCATE 6, 10: INPUT "Enter DATA File Name - "; DFILEN$
```

```
115 GOTO 600: REM *** Read DATA from DISC ***
120 LOCATE 7, 10: PRINT "Number of DATA Sets "; NCOL - 1
130 LOCATE 8, 10: PRINT "Number of DATA Points "; NROW
140 FOR D = 1 TO NCOL - 1
150 LOCATE 10, 10: PRINT "X-AXIS"
160 LOCATE 10, 10 + 10 * D: PRINT "DATA"; D
170 NEXT D
180 FOR I = 1 TO NROW
190 FOR J = 1 TO NCOL
200 IF I/11 = INT(I/11) AND J = 1 THEN 210 ELSE 250
210 FOR DELAY = 1 TO 5000: NEXT DELAY
220 FOR CROW = 1 TO 11
230 LOCATE 10 + CROW, 10: FOR SP = 1 TO 55: PRINT " "; : NEXT SP
240 NEXT CROW: GROW = 1
250 LOCATE 10 + GROW, 10 + (J - 1) * 10: PRINT Z(I, J)
260 NEXT J: GROW = GROW + 1
270 NEXT I
280 LOCATE 22, 10: PRINT "Press C to Continue "; : C$ = INPUT$(1)
290 IF INSTR("CC", C$) = 0 THEN 280
300 CLS : GOSUB 9000: EXT$ = ".DAT"
310 LOCATE 6, 10: PRINT "Please wait ..... saving DATA"
320 OPEN "O", #1, DDSK$ + DFILEN$ + EXT$
330 PRINT #1, NROW, NCOL
340 FOR I = 1 TO NROW
350 FOR J = 1 TO NCOL - 1
360 PRINT #1, USING "######.###"; Z(I, J);
370 NEXT J
380 PRINT #1, USING "######.###"; Z(I, NCOL)
390 NEXT I
500 CLOSE
510 OPEN "O", #2, DDSK$ + "IDFILE.PGD"
520 PRINT #2, DFILEN$
530 PRINT #2, EXT$
540 CLOSE #2
550 GOSUB 8500
560 IF EXT$ <> ".DAT" THEN 9600
570 GOTO 1100
600 LOCATE 10, 10:
601 PRINT "Is DATA stored on Drive "; DDSK$; " (Y/N)";
605 A$ = INPUT$(1)
610 IF A$ = "Y" OR A$ = "y" THEN 700
620 IF A$ = "N" OR A$ = "n" THEN 630 ELSE 600
630 LOCATE 10, 12:
631 PRINT "Enter New Logged DATA Drive (e.g. C:\PGDATA\)";
635 INPUT DDSK$
640 OPEN "O", #3, MDSK$ + PCF$
650 PRINT #3, SCR%
660 PRINT #3, SCD%
670 PRINT #3, MDSK$
```

```
680 PRINT #3, DDSK$
690 CLOSE #3
700 LOCATE 16, 10:
701 INPUT "Enter DATA File Extension (e.g. '.PRN')"; EXT$
710 LOCATE 18, 10:
711 PRINT "Please wait ..... reading DATA File "; DFILEN$
720 OPEN "I", #1, DDSK$ + DFILEN$ + EXT$
730 IF EXT$ = ".CON" THEN CLOSE : GOTO 500
740 IF EXT$ = ".EQD" THEN CLOSE : GOTO 500
750 INPUT #1, NROW, NCOL
760 ND = NCOL - 1
770 FOR I = 1 TO NROW
780 FOR J = 1 TO NCOL
790 INPUT #1, Z(I, J)
800 NEXT J
810 NEXT I
820 CLOSE #1
830 LOCATE 20, 10: PRINT "Do you wish to view DATA (Y/N) ";
840 A$ = INPUT$(1)
850 IF INSTR("YNyn", A$) = 0 THEN 830
860 IF A$ = "N" OR A$ = "n" THEN 300
870 CLS : GOSUB 9000
880 LOCATE 6, 10: PRINT "DATA File Name - "; DFILEN$
890 GOTO 120
899 REM *** JUMPS HERE IF DATA FILE NOT FOUND ***
900 CLS : GOSUB 9000
910 PRINT : FILES DDSK$ + "*" + EXT$
920 LOCATE 14, 5: PRINT "Please Type Correct DATA File Name"
930 LOCATE 16, 5:
931 INPUT "(Type 'QUIT' to Return to MENU) "; CNAME$
940 IF CNAME$ = "QUIT" OR CNAME$ = "quit" THEN 9600
950 GOTO 730
1000 CLS : GOSUB 9000: REM *** JUMPS HERE FROM ERROR IN 8500 ***
1010 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
1020 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
1030 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
1040 LOCATE 12, 10:
1041 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
1042 IF INSTR("HVhv", YLAB$) = 0 THEN 1040
1043 IF YLAB$ = "h" THEN YLAB$ = "H"
1044 IF YLAB$ = "v" THEN YLAB$ = "V"
1050 FOR I = 1 TO NCOL - 1
1060 LOCATE 12 + I, 10:
1061 PRINT "Enter Legend DATA "; I; " (Max 9) ";
1070 INPUT DAT$(I)
1080 NEXT I
1090 GOSUB 8400: REM Store Titles on Disc
1100 IF EXT$ <> ".DAT" THEN 9600
1110 CLS : GOSUB 9000
```

```
1120 LOCATE 6, 10: PRINT "Do you wish to print DATA (Y/N) ";
1130 A$ = INPUT$(1)
1140 IF INSTR("YNyn", A$) = 0 THEN 1130
1150 IF A$ = "N" OR A$ = "n" THEN 9600
```

"DISCDATA"—computer printout for data file 'PIDEFECT'

```
Presentation Graphics
Disc Data Input
DATA File - PIDEFECT
**********************
```

X-AXIS	DATA 1	DATA 2
1.000	37.000	67.000
2.000	36.000	3.000
3.000	18.000	17.000
4.000	9.000	13.000

Graph Title - Defects by construction activity

```
X-AXIS Title -
Y-AXIS Title -
```

DATA	Legend
1	Design
2	Specificatn
3	Workmanship
4	Supervision

APPENDIX A.27:
DATA GENERATION COMPUTER PROGRAM "KEYBEQUD"

This program was described in Section 1.10.3 and is used to generate a 3-D mathematical surface from a function *Z* in terms of *X* and *Y*.

Data generation

(a) Initialization and control Line numbers 10–90

This segment of code is similar to section (a) of Appendix A.25, the only exception being the size of the Z array which is changed to **Z(40,40)**.

(b) Data file generation Line numbers 100–540

Before running this program, either from GW-BASIC or QuickBASIC, the user must enter at Line 250 the function that is to be generated. For example, Eqt 1.1 from Section 1.10.3 would be entered as

Z(GROW,GCOL) = SIN(X^2 + Y^2) + SIN(Y)

The user is asked to enter from the keyboard the data file name and the limits of the surface plot (**XMIN, XMAX, YMIN** and **YMAX**). The *X*-dimension **(XD)** and *Y*-dimension **(YD)** of the plot is then split into 40 segments in each direction **XTC** and **YTC** to give the grid size. Once data generation is complete, the data is saved in a data file with a **'.EQD'** extension and as with the other two data entry programs, an **'IDFILE.PGD'** is saved for identification purposes.

(c) Text file entry Line numbers 1000–1100

This segment of code is similar to section (c) of Appendix A.25, but without Lines 1050–1080 as data series labels are not required. Note, only a vertical *Y*-axis label is possible at the side of the screen.
This program also requires the following routines.

(d) Appendix A.8	Line numbers 7000–7090
(e) Appendix A.2	Line numbers 8000–8180
(f) Appendix A.4	Line numbers 8400–8490
(g) Appendix A.15	Line numbers 9000–9190
(h) Appendix A.16	Line number 9600 only
(i) Appendix A.17	Line numbers 10000–10240

"KEYBEQUD"–BASIC program

```
10 REM <KEYBEQUD> Presentation Graphics DATA Generation Program
14 REM (C) Copyright P.H.Milne 1990 : Appendix A.27
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990
30 CLEAR : DROW = 40: DCOL = 40: DIM Z(DROW, DCOL): SR% = 2
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 CLS : PC$ = "Data File Generation": GOSUB 9000
110 LOCATE 6, 10:
111 INPUT "Enter DATA File Name - "; DFILEN$
120 LOCATE 8, 10:
121 INPUT "Enter X-Axis Limits (e.g. -2,2)" ; XMIN, XMAX
130 IF XMIN >= XMAX THEN 120
140 LOCATE 10, 10:
141 INPUT "Enter Y-Axis Limits (e.g. -2,2)" ; YMIN, YMAX
150 IF YMIN >= YMAX THEN 140
160 DX = (XMAX – XMIN): XTC = DX / DCOL
170 NCOL = (XMAX – XMIN) / XTC + 1
180 DY = (YMAX – YMIN): YTC = DY / DROW
190 NROW = (YMAX – YMIN) / YTC + 1
200 LOCATE 20, 10:
201 PRINT "Please Wait ..... Generating Data File"
```

```
210 Y = YMAX
220 FOR GROW = 0 TO NROW - 1
230 X = XMIN
240 FOR GCOL = 0 TO NCOL
250 Z(GROW, GCOL) = SIN(X ^ 2 + Y ^ 2) + SIN(Y)
260 X = X + XTC
270 NEXT GCOL
280 Y = Y - YTC
290 NEXT GROW
300 CLS: PF$ = DFILEN$: GOSUB 9000: GOSUB 7000
310 LOCATE 15, 10: PRINT "Please wait ..... saving DATA"
320 OPEN "O", #1, DDSK$ + DFILEN$ + ".EQD"
330 PRINT #1, NROW, NCOL
340 PRINT #1, ZMAX, ZMIN
350 PRINT #1, XMIN, XMAX, YMIN, YMAX
360 FOR I = 0 TO NROW - 1
370 FOR J = 0 TO NCOL - 1
380 IF J = NCOL - 1 THEN 410
390 PRINT #1, USING "######.###"; Z(I, J) ;
400 GOTO 420
410 PRINT #1, USING "######.###"; Z(I, J)
420 NEXT J
430 NEXT I
500 CLOSE
510 OPEN "O", #2, DDSK$ + "IDFILE.PGD"
520 PRINT #2, DFILEN$
530 PRINT #2, ".EQD"
540 CLOSE #2
1000 CLS : GOSUB 9000
1010 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
1020 LOCATE 8, 10: INPUT "Enter Second Title (Max 40) "; LXAXI$
1030 LOCATE 10, 10: INPUT "Enter Vertical Title (Max 11) "; LYAXI$
1040 YLAB$ = "V"
1090 GOSUB 8400
1100 GOTO 9600
```

APPENDIX A.28:
HPGL GRAPHICS TEMPLATE COMPUTER ROUTINE AT LINE 7300

This plotter routine was described in Section 1.11 and replaces the screen graphics template described in Section 1.9.9.

HPGL graphics template—BASIC routine

```
7300 GOSUB 50000
7305 PRINT #3, "IN;"
7310 PRINT #3, "IP"; IPXMIN; "," ; IPYMIN; "," ; IPXMAX; "," ;
   IPYMAX; ";"
7315 PRINT #3, "SC 0,640,0,200;"
```

```
7320 PRINT #3, "SP2;PU;PA 10,20;PD;PA 620,20;": GOSUB 60000
7325 PRINT #3, "PA 620,180;": GOSUB 60000
7330 PRINT #3, "PA 10,180;": GOSUB 60000
7335 PRINT #3, "PA 10,20;PU;": GOSUB 60000
7350 IF GRAPH = 0 THEN RETURN
7360 PRINT #3, "SP1;PA"; X0; ","; Y0 + YM;"; ";": GOSUB 60000
7365 PRINT #3, "PD;PA "; X0; ","; Y0; ";": GOSUB 60000
7370 PRINT #3, "PA "; X0 + XM; ","; Y0; ";": GOSUB 60000
7380 IF GRAPH = 1 THEN RETURN
7385 PRINT #3, "SI0.2,0.3;"
7390 FOR LXD = 0 TO 10
7400 AVGX = X0 + LXD * 50: AVGY = Y0 - 11: LDAT = XMIN + LXD * XTC
7410 PRINT #3, "PU "; AVGX; ","; Y0; ";XT;": GOSUB 60000
7420 PRINT #3, "PU "; AVGX; ","; Y0; ";CP-2, -1; LB"; LDAT; CHR$(3)
7430 NEXT LXD
7440 FOR LYD = 0 TO 10
7450 AVGY = Y0 + LYD * 12: AVGX = X0 - 32: LDAT = YMIN + LYD * YTC
7460 PRINT #3, "PU"; X0; ","; AVGY; ";YT;": GOSUB 60000
7470 PRINT #3, "PU"; X0; ","; AVGY; ";CP-4.5,-0.25;LB"; LDAT; CHR$
   (3)
7480 NEXT LYD
7490 RETURN
```

APPENDIX A.29:
HPGL TEXT LABELS COMPUTER ROUTINE AT LINE 7500

This plotter routine was described in Section 1.11 and replaces the screen graphics text labels described in Section 1.9.10.

HPGL text labels—BASIC routine

```
7500 LNT = LEN(TITLE$)
7501 PRINT #3, "SP2;SI0.3,0.55;SL0.2;": GOSUB 60000
7502 PRINT #3, "PU;PA320,185;": GOSUB 60000
7503 PRINT #3, "CP"; -LNT/2; "0;": GOSUB 60000
7510 PRINT #3, "LB"; TITLE$; CHR$(3): GOSUB 60000
7511 PRINT #3, "SP1;SL;SI0.2,0.3;PU;PA320,35;": GOSUB 60000
7520 LNT = LEN(LXAXI$): PRINT #3, "CP"; -LNT / 2; "0;": GOSUB 60000
7530 PRINT #3, "LB"; LXAXI$; CHR$(3): GOSUB 60000
7570 PRINT #3, "PU;PA30,75;": GOSUB 60000
7575 PRINT #3, "DI 0,1;": GOSUB 60000
7580 PRINT #3, "LB"; LYAXI$; CHR$(3): GOSUB 60000
7585 PRINT #3, "DI 1,0;"
7590 RETURN
```

APPENDIX A.30:
HPGL BACKGROUND GRID COMPUTER ROUTINE AT LINE 7600

This plotter routine was described in Section 1.11 and replaces the screen graphics background grid described in Section 1.9.11.

HPGL background grid—BASIC routine

```
7600 PRINT #3, "PU;SP1;": GOSUB 60000
7610 PRINT #3, "LT2, 1;"
7620 FOR LYD = 1 TO 10
7630 AVGY = Y0 + LYD * 12
7640 PRINT #3, "PU;PA"; X0; ","; AVGY; ";": GOSUB 60000
7645 PRINT #3, "PD;PA"; X0 + XM; ","; AVGY; ";": GOSUB 60000
7650 NEXT LYD
7660 IF VGRID = 0 THEN 7705
7670 FOR LXD = 1 TO 10
7680 AVGX = X0 + LXD * 50
7690 PRINT #3, "PU;PA"; AVGX; ","; YO; ";": GOSUB 60000
7695 PRINT #3, "PD;PA"; AVGX; ","; YO + YM; ";": GOSUB 60000
7700 NEXT LXD
7705 PRINT #3, "LT;"
7710 RETURN
```

APPENDIX A.31:
HPGL BAR CHART TEMPLATE COMPUTER ROUTINE AT LINE 7800

This plotter routine was described in Section 1.11 and replaces the screen graphics bar chart template described in Section 1.9.12.

HPGL bar chart template—BASIC routine

```
7800 GOSUB 50000
7805 PRINT #3, "IN;"
7810 PRINT #3, "IP"; IPXMIN; ","; IPYMIN; ","; IPXMAX; ","; IPYMAX;
   ";"
7815 PRINT #3, "SC 0,640,0,200;"
7820 PRINT #3, "SP2;PU;PA 10,20;PD;PA 620,20;": GOSUB 60000
7825 PRINT #3, "PA 620,180;": GOSUB 60000
7830 PRINT #3, "PA 10,180;": GOSUB 60000
7835 PRINT #3, "PA 10,20;PU;": GOSUB 60000
7850 IF GRAPH = 0 THEN RETURN
7860 PRINT #3, "SP1;PA"; X0; ","; Y0 + YM; ";": GOSUB 60000
7865 PRINT #3, "PD;PA "; X0; ","; Y0; ";": GOSUB 60000
7870 PRINT #3, "PA "; X0 + XM; ","; Y0; ";": GOSUB 60000
7880 IF GRAPH = 1 THEN RETURN
7885 PRINT #3, "SI0.2,0.3;"
7890 FOR LYD = 1 TO NROW
7900 AVGY = Y0 + INT(YTC / 2) + (LYD - 1) * YTC: AVGX = X0 - 11
```

```
7905 LDAT = Z(LYD, 1)
7910 PRINT #3, "PU "; X0; ","; AVGY; ";YT;": GOSUB 60000
7920 PRINT #3, "PU "; X0; ","; AVGY; ";CP-4.5,-0.25;LB"; LDAT; CHR$
    (3)
7930 NEXT LYD
7940 FOR LXD = 0 TO 10
7950 AVGX = X0 + LXD * 50: AVGY = Y0 - 3: LDAT = YMIN + LXD * XTC
7960 PRINT #3, "PU "; AVGX; ","; Y0; ";XT;": GOSUB 60000
7970 PRINT #3, "PU "; AVGX; ","; Y0; CP-2,-1;LB"; LDAT; CHR$(3)
7980 NEXT LXD
7990 RETURN
```

APPENDIX A.32:

HPGL DATA SYMBOL DISPLAY COMPUTER ROUTINE AT LINE 40000

This routine was described in Section 1.11 and replaces the screen graphics data symbol display described in Section 1.9.17.

HPGL data symbol display—BASIC routine

```
39999 REM ***** Symbols for Graphs *****
40000 W = 3: H = 1.5
40010 ON J GOTO 40100, 40200, 40300, 40400, 40500
40099 REM ***** 1 - Square Box *****
40100 PRINT 13, "PU;SP2;": GOSUB 60000
40110 PRINT #3, "PU;PA"; AVGX - W; ","; AVGY - H;";"
40115 PRINT #3, "EA"; AVGX + W; " , " ; AVGY + H; ";": GOSUB 60000
40120 RETURN
40199 REM ***** 2 - Triangle *****
40200 PRINT #3, "PU;SP4 ": GOSUB 60000
40210 PRINT #3, "PU;PA" AVGX; " , " ; AVGY + H;";"
40220 PRINT #3, "PD;PA" AVGX - W; ","; AVGY - H; ";"
40230 PRINT #3, "PA"; AVGX + W; ","; AVGY - H; ";"
40240 PRINT #3, "PA"; AVGX; ","; AVGY + H; " ; " : GOSUB 60000
40250 RETURN
40299 REM ***** 3 - Diamond *****
40300 PRINT #3, "PU;SP1;": GOSUB 60000
40310 PRINT #3, "PU;PA"; AVGX; ","; AVGY + H; "."
40320 PRINT #3, "PD;PA"; AVGX - W; ","; AVGY; ";"
40325 PRINT #3, "PA"; AVGX; ","; AVGY - H; ";"
40330 PRINT #3, "PA"; AVGX + W; ","; AVGY; " ; "
40340 PRINT #3, "PA"; AVGX; ","; AVGY + H; ";": GOSUB 60000
40350 RETURN
40399 REM ***** 4 - Cross 'x' sign *****
40400 PRINT #3, "PU;SP4;": GOSUB 60000
40410 PRINT #3, "PU;PA"; AVGX - W; ","; AVGY - H; ";"
40420 PRINT #3, "PD;PA"; AVGX + W; ","; AVGY + H; ";"
40430 PRINT #3, "PU;PA"; AVGX - W; ","; AVGY + H; ";"
40440 PRINT #3, "PD;PA"; AVGX + W; ","; AVGY - H; ";": GOSUB 60000
```

```
40450 RETURN
40499 REM ***** 5 - Plus '+' sign *****
40500 PRINT #3, "PU;SP3;": GOSUB 60000
40510 PRINT #3, "PU;PA"; AVGX; ","; AVGY - H; ";"
40520 PRINT #3, "PD,PA"; AVGX; ","; AVGY + H; ";"
40530 PRINT #3, "PU;PA"; AVGX - W; " , " ; AVGY; ";"
40540 PRINT #3,"PD;PA"; AVGX + W; " , " ; AVGY;";": GOSUB 60000
40550 RETURN
```

Appendix B
Line graphs and area charts

APPENDIX B.1:
LINE GRAPHS COMPUTER PROGRAM "GRAPH-B1"

This program was discussed in Section 2.2.1 and plots a zigzag line through values of *Y* graphed against an increasing *X* value.

Zigzag line graph

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables which require to be initialized for use in the program. The array **Z(100,6)** is set for a maximum of 100 rows of data containing 5 data sets. The colour array **CL%(5)** is set to contain five different colours, one for each data set.

(b) Data loading Line numbers 100–180

All the graph programs automatically run the routine described in Section 1.9.2 **(GOSUB 8200)** to recall the last data file stored in the identity file **'IDFILE.PGD'**. This data file can be loaded or another data file recalled using the routine at Line 8800. If a monochrome screen is being used, then tile patterns are also loaded using the routine at Line 8600.

(c) Data check Line numbers 200–440

This segment of code loads the routine at Line 7000 to check on the maximum and minimum data values for the graph. The user can alter these if required and enter new values for the graph display. The text description file is then read using the routine at Line 8500, and this can also be changed if the user wishes.

(d) Text file entry Line numbers 500–590

This routine is only used if no text file is found in the previous section (c), or if the user wishes to change the title and graph axes labels.

(e) Graph presentation selection Line numbers 600–750

The user now has to choose how the graph is to be displayed on screen, with five different options available, as discussed in Section 2.2.1. Depending on the selection **(NSEL),** four numeric variables are set to **1** if required, and to **0** if not required. These are **GLIN** for lines, **MARK** for markers, **HGRID** for horizontal grid and **VGRID** for vertical grid.

(f) Zigzag line graph display

Line numbers 1000–1380

This is the main part of the program and uses several of the graphics presentation routines described in Chapter 1, to set up the screen **(GOSUB 7300),** title the graph **(GOSUB 7500),** draw a grid if required **(GOSUB 7600),** and then plot either the data point markers or the lines, or both. For multiple zigzag line graphs, different colours and line styles **(ST)** are used to display each different set of data. The user can obtain a hardcopy printout by pressing the [Shift] and [Prt sc] keys simultaneously, before continuing.
This program also requires the following routines.

(g) Appendix A.8	Line numbers 7000–7090
(h) Appendix A.9	Line numbers 7100–7290
(i) Appendix A.10	Line numbers 7300–7490
(j) Appendix A.11	Line numbers 7500–7590
(k) Appendix A.12	Line numbers 7600–7710
(l) Appendix A.2	Line numbers 8000–8180
(m) Appendix A.3	Line numbers 8200–8390
(n) Appendix A.4	Line numbers 8400–8490
(o) Appendix A.5	Line numbers 8500–8590
(p) Appendix A.14	Line numbers 8600–8660
(q) Appendix A.3	Line numbers 8800–8990
(r) Appendix A.15	Line numbers 9000–9050
(s) Appendix A.16	Line numbers 9500–9600
(t) Appendix A.17	Line numbers 10000–10310
(u) Appendix A.18	Line numbers 40000–40590
(v) Appendix A.19	Line numbers 55000–56960

“GRAPH-B1”—BASIC program

```
10 REM <GRAPH-B1> Presentation Graphics Program B.1 - Line Graphs
14 REM (C) Copyright P.H.Milne 1990
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: *** Max 100 Rows of Data ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
70 X0 = 70: Y0 = 50: A$ = "*": DATBX = 0
80 DIM Z(100, 6), CL%(5)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Line Graph"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
```

```
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 140
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N) ";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 260
290 IF A$ = "N" OR A$ = "n" THEN 340
300 LOCATE 17, 10: INPUT "New X-AXIS (X-min, X-max) "; XMIN, XMAX
310 IF XMIN >= XMAX THEN 300
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "; YMIN, YMAX
330 IF YMIN >= YMAX THEN 320
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 500
370 YTC = (YMAX - YMIN) / 10
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000:
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 LOCATE 12, 10:
541 INPUT "Y-AXIS Title (H)oriz, or (V)ert. "; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
544 IF YLAB$ = "v" THEN YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA"; I; "(Max 9)";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS : GOSUB 9000
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
```

```
620 LOCATE 8, 10: PRINT "<1> Line Chart with Lines and Markers"
630 LOCATE 10, 10: PRINT "<2> Lines only"
640 LOCATE 12, 10: PRINT "<3> Markers only"
650 LOCATE 14, 10: PRINT "<4> Lines and Markers with Horizontal
   Grid"
660 LOCATE 16, 10:
665 PRINT "<5> Lines and Markers with Horizontal & Vertical Grid"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12345", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740, 750
710 GLIN = 1: MARK = 1: HGRID = 0: VGRID = 0: GOTO 1000
720 GLIN = 1: MARK = 0: HGRID = 0: VGRID = 0: GOTO 1000
730 GLIN = 0: MARK = 1: HGRID = 0: VGRID = 0: GOTO 1000
740 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 1000
750 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 1: YLAB$ = "V": GOTO 1000
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 FOR J = 1 TO NCOL - 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1060 NEXT J
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 1200
1110 FOR J = 1 TO NCOL - 1
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 NEXT J
1180 IF GLIN = 0 THEN 1380
1200 FOR J = 1 TO NCOL - 1
1210 FOR K = 1 TO NROW - 1
1220 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1230 Y1 = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1240 X2 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0
1250 Y2 = (Z(K + 1, J + 1) - YMIN) * 12 / YTC + Y0
1260 ON J GOTO 1270, 1280, 1290, 1300, 1310
1270 ST = &HFFFF: CL% = 4: GOTO 1320
1280 ST = &HCCCC: CL% = 1: GOTO 1320
1290 ST = &HF0FF: CL% = 5: GOTO 1320
1300 ST = &HCCFF: CL% = 3: GOTO 1320
1310 ST = &HCCFC: CL% = 2
1320 IF X1 > 560 THEN 1360
```

```
1330 IF Y1 > 160 THEN 1360
1340 PSET (X1, Y1)
1350 LINE - (X2, Y2), CL%, , ST
1360 NEXT K
1370 NEXT J
1380 GOTO 9500
```

COMPUTER PROGRAM "GPLOT-B1"

To convert the program **"GRAPH-B1"** from screen graphics to an HPGL plotter the following changes are required:

replace segment (f) with the code for Lines 1000–1380, which are given at the end of this section
replace segment (i) with Appendix A.28
replace segment (j) with Appendix A.29
replace segment (k) with Appendix A.30
replace segment (u) with Appendix A.32
delete segments (p) and (v) and add the following:

(aa) Appendix A.19	Line numbers 50000–50180
(bb) Appendix A.20	Line numbers 50500–50570
(cc) Appendix A.21	Line numbers 51000–51060
(dd) Appendix A.24	Line numbers 60000–60150

"GPLOT-B1"—BASIC program—segment (f) only

```
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1005 GOSUB 50500: REM Header
1010 GOSUB 7500: REM Titles
1020 FOR J = 1 TO NCOL - 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 23
1040 GOSUB 40000
1045 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY - 2; ";": GOSUB 60000
1050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1060 NEXT J
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 1200
1110 FOR J = 1 TO NCOL - 1
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 NEXT J
1180 IF GLIN = 0 THEN 1380
1200 FOR J = 1 TO NCOL - 1
1210 FOR K = 1 TO NROW - 1
```

```
1220 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1230 Y1 = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1240 X2 = (Z(K + 1,1) - XMIN) * 50 / XTC + X0
1250 Y2 = (Z(K + 1, J + 1) - YMIN) * 12 / YTC + Y0
1320 IF X1 > 570 THEN 1360
1330 IF Y1 > 170 THEN 1360
1350 GOSUB 51000
1360 NEXT K
1370 NEXT J
1375 PRINT #3, "PU; PA0, 0; SP0;": CLOSE #3
1380 GOTO 9530
```

APPENDIX B.2:
LINEAR REGRESSION COMPUTER PROGRAM “GRAPH-B2”

This program was discussed in Section 2.2.2 and plots a trend line graph using linear regression.

Linear regression

(a) Initialization and control Line numbers 10–90

See notes for section (a) of Appendix B.1.

(b) Data loading Line numbers 100–180

See notes for section (b) of Appendix B.1.

(c) Data check Line numbers 200–440

See notes for section (c) of Appendix B.1.

(d) Text file entry Line numbers 500–590

See notes for section (d) of Appendix B.1.

(e) Graph presentation selection Line numbers 600–750

See notes for section (e) of Appendix B.1.

(f) Data set selection Line numbers 800–850

If the data file contains only one data set, the program resumes at Line 1000. However, if the data file contains multiple data sets, this routine allows the user to choose which data set to display.

(g) Line graph display Line numbers 1000–1170

This routine is described in section (f) of Appendix B.1, with the exception that only one data set is now displayed, and hence the **REM** lines indicating previous ‘for…next’ loops.

(h) Linear regression trend line Line numbers 2000–2300

This segment of code now utilizes the least squares equations, Eqts 2.1–2.10, described in Section 2.2.2, plots the trend line through the data and displays the solution to Eqt. 2.1 on screen. The two variables *a* and *b* are then stored on disc in a data file with a **‘.LIN’** extension for future plotting.

(i) Data printout? Line numbers 4000–4070

At the completion of the plot, the user can obtain a hardcopy printout by pressing the [Shift] and [Prt sc] keys simultaneously. The user also has an opportunity to obtain a printout of the data, giving the various statistical parameters and any residuals or variance with the actual data, by using the routine at Line 6500.

This program also requires the following routines.

(j) Appendix A.7	Line numbers 6500–6810
(k) Appendix A.8	Line numbers 7000–7090
(I) Appendix A.9	Line numbers 7100–7290
(m) Appendix A.10	Line numbers 7300–7490
(n) Appendix A.11	Line numbers 7500–7580
(o) Appendix A.12	Line numbers 7600–7710
(p) Appendix A.2	Line numbers 8000–8180
(q) Appendix A.3	Line numbers 8200–8390
(r) Appendix A.4	Line numbers 8400–8480
(s) Appendix A.5	Line numbers 8500–8580
(t) Appendix A.14	Line numbers 8600–8660
(u) Appendix A.3	Line numbers 8800–8990
(v) Appendix A.15	Line numbers 9000–9050
(w) Appendix A.16	Line numbers 9500–9600
(x) Appendix A.17	Line numbers 10000–10310
(y) Appendix A.18	Line numbers 40000–40590
(z) Appendix A.19	Line numbers 55000–56960

"GRAPH-B2"—BASIC program

```
10 REM <GRAPH-B2> Presentation Graphics Program B.2 - Line Graphs
14 REM (C) Copyright P.H.Milne 1990 *** - Linear Regression ***
16 REM ALL RIGHTS RESERVED *** - 100 DATA Points ***
20 REM VERSION PC-1.00, 22/05/90
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 DIM Z(100, 6), CL%(5)
70 X0 = 70: Y0 = 50: A$ = "*": DATBX = 0
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Linear Regression"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 150
170 IF A$ = "N" OR A$ = "n" THEN 9500
```

```
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N) ";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 270
290 IF A$ = "N" OR A$ = "n" THEN 340
300 LOCATE 17, 10: INPUT "New X-AXIS (X-min, X-max) "; XMIN, XMAX
310 IF XMIN >= XMAX THEN 300
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "; YMIN, YMAX
330 IF YMIN >= YMAX THEN 320
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 500
370 YTC = (YMAX - YMIN) / 10
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 420
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 LOCATE 12, 10:
541 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
544 IF YLAB$ = "v" THEN YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS : GOSUB 9000: J = 1
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 8, 10: PRINT "<1> Line Chart with Lines and Markers"
630 LOCATE 10, 10: PRINT "<2> Lines only"
640 LOCATE 12, 10: PRINT "<3> Markers only"
650 LOCATE 14, 10: PRINT "<4> Lines and Markers with Horizontal
   Grid"
```

```
660 LOCATE 16, 10:
665 PRINT "<5> Lines and Markers with Horizontal & Vertical Grid"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12345", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740, 750
710 GLIN = 1: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
720 GLIN = 1: MARK = 0: HGRID = 0: VGRID = 0: GOTO 800
730 GLIN = 0: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
740 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 800
750 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 1: YLAB$ = "V": GOTO 800
800 CLS : GOSUB 9000: IF ND = 1 THEN 1000
810 LOCATE 10, 10: PRINT "Select DATA Set 1 -"; ND
820 LOCATE 12, 10: PRINT "Enter No."; : DAT$ = INPUT$(1)
830 IF INSTR("12345", DAT$) = 0 THEN 800
840 J = VAL(DAT$)
850 CLS : GOSUB 9000
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 REM
1030 AVGX = 94 + 103 * (1 - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + 1 * 13: PRINT DAT$(J)
1060 REM
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 2000
1110 REM
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 REM
1999 REM ***** Least Squares *****
2000 SUMX = 0: SUMY = 0: SUMXY = 0: SUMXX = 0: SUMYY = 0
2010 FOR I = 1 TO NROW
2020 SUMX = SUMX + Z(I, 1)
2030 SUMY = SUMY + Z(I, J + 1)
2040 SUMXY = SUMXY + Z(I, 1) * Z(I, J + 1)
2050 SUMXX = SUMXX + Z(I, 1) ^ 2
2060 SUMYY = SUMYY + Z(I, J + 1) ^ 2
2070 NEXT I
2080 FACTB = (NROW * SUMXY - SUMX * SUMY) / (NROW * SUMXX - SUMX ^
   2)
2090 XBAR = SUMX / NROW: YEAR = SUMY / NROW
2100 FACTA = (YEAR - FACTB * XBAR)
```

```
2110 YP1 FACTA + FACTB * XMIN
2120 YP2 FACTB * XMAX
2130 SCAX = 500 / (10 * XTC): SCAY = 120 / (10 * YTC)
2140 IF SCR% = 2 THEN RGB% = 12
2150 X1 = X0: Y1 = Y0 + SCAY * (YP1 - YMIN)
2160 X2 = X0 + (XMAX - XMIN) * SCAX: Y2 = Y0 + SCAY * (YP2 - YMIN)
2170 LINE (X1, Y1) - (X2, Y2), RGB%
2180 A = INT(FACTA * 1000 + .1) / 1000
2190 B = INT(FACTB * 1000 + .1) / 1000
2200 LOCATE 22, 25: PRINT "EQN. Y = ";A;" + (";B;") * X"
2210 CLOSE : OPEN "O", #3, DDSK$ + DFILEN$ + ".LIN"
2220 PRINT #3, 1
2230 PRINT #3, FACTA
2240 PRINT #3, FACTB
2250 CLOSE
2260 STDX = SQR((SUMXX - NROW * XBAR ^ 2) / (NROW - 1))
2270 STDY = SQR((SUMYY - NROW * YBAR ^ 2) / (NROW - 1))
2280 RT = (SUMXY - SUMX * SUMY / NROW) / (NROW - 1)
2290 RB = STDX * STDY
2300 RXY = RT / RB
4000 A$ = INPUT$(1)
4010 IF A$ = "" THEN 4000 ELSE 4020
4020 LOCATE 24, 10: PRINT "Do you wish Printout of Data (Y/N)";
4030 A$ = INPUT$(1)
4040 IF INSTR("YNyn", A$) = 0 THEN 4030
4050 IF A$ = "Y" OR A$ = "y" THEN 6500
4060 LOCATE 24, 10: PRINT " ";
4070 GOTO 9530
```

"GRAPH-B2"—computer printout for data file 'TCELSIUS'

```
Linear Regression
DATA File : TCELSIUS
Temperature - Celsius to Fahrenheit
********************************************
Factor A = 32
Factor B = 1.8
Mean X = -15
Mean Y = 5
Standard Deviation X = 62.04837
Standard Deviation Y = 111.6871
Coefficient of Correlation = 1
Determination Coefficient = 1
```

X-Data	Y-Data	Y-Calc	Residual
• 130.000	• 202.000	• 202.000	0.000
• 40.000	• 40.000	• 40.000	0.000
5.000	41.000	41.000	0.000

X-Data	Y-Data	Y-Calc	Residual
15.000	59.000	59.000	0.000
25.000	77.000	77.000	0.000
35.000	95.000	95.000	0.000

COMPUTER PROGRAM "GPLOT-B2"

To convert the programs **"GRAPH-B2"** and **"GRAPH-B3"** from screen graphics to an HPGL plotter the following changes are required:

add Lines 111–113 to segment (b) as shown at the end of this section
replace segment (g) with the code for Lines 1000–1200, which are given at the end of this section
replace segment (h) with the code for Lines 2000–3020, which are given at the end of this section
replace segment (m) with Appendix A.28
replace segment (n) with Appendix A.29
replace segment (o) with Appendix A.30
alter Lines 8270–8290 in segment (q) as shown in the code at the end of this section
replace segment (y) with Appendix A.32
delete segments (i), (j), (t) and (z)
add segments (aa)–(dd) from **"GPLOT-B1"**

"GPLOT-B2"—BASIC program—Lines 111–113, segments (g) and (h), and part (q) only

```
111 IF TSEL = 1 THEN PC$ = "Linear Regression"
112 IF TSEL = 2 THEN PC$ = "Exponential Curve"
113 IF TSEL = 3 THEN PC$ = "Power Curve"

1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1005 GOSUB 50500: REM Header
1010 GOSUB 7500: REM Titles
1020 REM
1030 AVGX = 94 + 103 * (1 - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1045 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY - 2; ";": GOSUB 60000
1050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1060 REM
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 1200
1110 REM
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
```

```
1160 NEXT K
1170 REM
1200 ON TSEL GOTO 2000, 2500, 2500
2000 REM ***** Least Squares *****
2130 SCAX = 500 / (10 * XTC): SCAY = 120 / (10 * YTC)
2150 X1 = X0: Y1 = Y0 + SCAY * (YP1 - YMIN)
2160 X2 = X0 + (XMAX - XMIN) * SCAX: Y2 = Y0 + SCAY * (YP2 - YMIN)
2170 PRINT #3, "SP2;PU;PA"; X1; ","; Y1; ";": GOSUB 60000
2180 PRINT #3, "PD;PA"; X2; ","; Y2; ";": GOSUB 60000
2190 PRINT #3, "SI0.2,0.3;PU;PA200,25;": GOSUB 60000
2200 PRINT #3, "LB EQN. Y = "; FACTA; " + ("; FACTB; ") * X" ;
2205 PRINT #3, CHR$(3): GOSUB 60000
2310 PRINT #3, "PU;PA0,0;SP0;": CLOSE #3
2320 GOTO 9530
2500 REM **** Exponential & Power Curve Fit ****
2510 RNG = XMAX - XMIN
2520 PRINT #3, "SI0.2,0.3;PU;PA200,25;": GOSUB 60000
2690 ON TSEL - 1 GOTO 2700, 2750
2700 PRINT #3, "LB EQN. Y = "; FACTA; "* e ^ ("; FACTB; " * X)";
2705 PRINT #3, CHR$(3): GOSUB 60000
2740 GOTO 2790
2750 PRINT #3, "LB EQN. Y = "; FACTA; "* X ^ ("; FACTB; ")";
2755 PRINT #3, CHR$(3): GOSUB 60000
2790 FOR P = XMIN TO XMAX STEP RNG / 50
2795 Q = P + RNG / 100
2800 X1 = (P - XMIN) * 50 / XTC + X0: IF X1 > 570 THEN 2870
2805 X2 = (Q - XMIN) * 50 / XTC + X0
2810 IF TSEL = 3 THEN 2830
2820 Y1 = (FACTA * EXP(FACTB * P)) * 12 / YTC + Y0
2825 Y2 = (FACTA * EXP(FACTB * Q)) * 12 / YTC + Y0: GOTO 2840
2830 Y1 = (FACTA * P ^ FACTB) * 12 / YTC + Y0
2835 Y1 = (FACTA * Q ^ FACTB) * 12 / YTC + Y0
2840 IF Y1 > 170 THEN 2870
2850 PRINT #3, "SP2;PU;PA"; X1; ","; Y1; ";": GOSUB 60000
2860 PRINT #3, "PD;PA"; X2;","; Y2; ";": GOSUB 60000
2870 NEXT P
3010 PRINT #3, "PU;PA0,0;SP0;": CLOSE #3
3020 GOTO 9530

8270 IF INSTR("YNyn", A$) = 0 THEN 8260
8275 IF A$ = "N" OR A$ = "n" THEN 8800
8280 OPEN "I", #2, DDSK$ + DFILEN$ + ".LIN"
8281 INPUT #2, TSEL
8282 INPUT #2, FACTA
8283 INPUT #2, FACTB
8284 CLOSE : GOTO 8290
8285 LOCATE 10, 10: PRINT "You must prepare Trend Line Data first"
8286 GOTO 9530
8290 LOCATE 10, 10: PRINT "Please wait ..... reading DATA"
```

APPENDIX B.3:
EXPONENTIAL AND POWER CURVE FIT COMPUTER PROGRAM "GRAPH-B3"

These programs were discussed in Section 2.2.2 to fit either an exponential curve through the data points using Eqt. 2.11, or a power curve through the data points using Eqt. 2.12. These two curve fit programs are very similar, so will be discussed together. Note the plotter output is contained in program **"GPLOT-B2"**.

Exponential and power curve fit

(a) Initialization and control Line numbers 10–90

See notes for section (a) of Appendix B.1. An additional array **M(8)** is used to store statistical parameters.

(b) Data loading Line numbers 100–180

See notes for section (b) of Appendix B.1.

(c) Data check Line numbers 200–440

See notes for section (c) of Appendix B.1.

(d) Text file entry Line numbers 500–580

See notes for section (d) of Appendix B.1.

(e) Graph presentation selection Line numbers 600–750

See notes for section (e) of Appendix B.1.

(f) Curve fit selection **L**ine numbers 800–890

This menu gives the user a choice of fitting, either an exponential curve or a power curve through the data points. Note, this program is for a single data set. If the data file contains more than one data set, it will only fit a curve through the first data set.

(g) Line graph display Line numbers 1000–1180

See notes for section (g) in Appendix B.2.

(h) Exponential and power curve fit Line numbers 2500–3010

This segment of code now utilizes the equations described in Section 2.2.2 to fit either an exponential curve or a power curve through the data set. The solution to Eqt. 2.11 or 2.12 is presented on screen with the curve. This routine also calculates all the statistical parameters, including the correlation coefficient and coefficient of determination.

(i) Data printout? Line numbers 4000–4090

See the notes for section (i) of Appendix B.2.
This program also requires the routines (j) to (z) listed for Appendix B.2.

"GRAPH-B3"—BASIC program

```
10 REM <GRAPH-B3> Presentation Graphics Program B.3 - Line Graphs
14 REM (C) Copyright P.H.Milne 1990 *** - Exponential ***
16 REM ALL RIGHTS RESERVED *** - Power Curve ***
20 REM VERSION PC-1.00, 22/05/90 :*** Maximum 100 Rows of Data ***
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
70 X0 = 70: Y0 = 50: A$ = "*": DATBX = 0
80 DIM Z(100, 2), CL%(5), M(8)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Curve Fitting"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 150
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N) ";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 260
290 IF A$ = "N" OR A$ = "n" THEN 340
300 LOCATE 17, 10: INPUT "New X-AXIS (X-min, X-max) "; XMIN, XMAX
310 IF XMIN >= XMAX THEN 300
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "; YMIN, YMAX
330 IF YMIN >= YMAX THEN 320
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 500
370 YTC = (YMAX - YMIN) / 10
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 420
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000:
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
```

```
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 LOCATE 12, 10:
541 INPUT "Y-AXIS Title (H)oriz. or (V)ert."; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
544 IF YLAB$ = "v" THEN YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS : GOSUB 9000
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 8, 10: PRINT "<1> Curve Fit with Data Points"
630 LOCATE 10, 10: PRINT "<2> Curve Fit only"
640 LOCATE 12, 10: PRINT "<3> Data Points only"
650 LOCATE 14, 10: PRINT "<4> Curve and Data with Horizontal Grid"
660 LOCATE 16, 10:
665 PRINT "<5> Curve and Data with Horizontal & Vertical Grid"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12345", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740, 750
710 GLIN = 1: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
720 GLIN = 1: MARK = 0: HGRID = 0: VGRID = 0: GOTO 800
730 GLIN = 0: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
740 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 800
750 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 1: YLAB$ = "V": GOTO 800
800 CLS : GOSUB 9000
810 LOCATE 6, 10: PRINT "Select Type of Graph :-"
820 LOCATE 10, 10: PRINT "<1> Exponential Curve Fit"
830 LOCATE 12, 10: PRINT "<2> Power Curve Fit"
840 LOCATE 14, 10: PRINT "<3> Quit - Return to Master Menu"
850 LOCATE 20, 10: PRINT "Enter Choice "; : TYP$ = INPUT$(1)
860 IF INSTR("12345" TYP$) = 0 THEN 850
870 NTYP = VAL(TYP$)
880 IF NTYP = 3 THEN 9600
890 DIM RX(NROW), RY(NROW), RX2(NROW), RY2(NROW)
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 J = 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1060 REM
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 2500
```

```
1110 J = 1
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 REM
1180 GOTO 2500
2499 REM **** Exponential & Power Curve Fit ****
2500 FOR K = 1 TO 8
2510 M(K) = 0
2520 NEXT K
2530 FOR J = 1 TO NROW
2540 SUMX = SUMX + Z(J, 1): REM SIGMA X
2550 SUMXX = SUMXX + Z(J, 1) ^ 2: REM SIGMA X^2
2560 SUMLY = SUMLY + LOG(Z(J, 2)): REM SIGMA LOG(Y)
2570 SUMYY = SUMYY + Z(J, 2) ^ 2: REM SIGMA Y^2
2580 SUMXLY = SUMXLY + Z(J, 1) * LOG(Z(J, 2)): REM SIGMA X*LOG(Y)
2590 SUMLX = SUMLX + LOG(Z(J, 1)): REM SIGMA LOG(X)
2600 SUMLXY = SUMLXY + LOG(Z(J, 1)) * LOG(Z(J,2)): REM LOG(X*Y)
2610 SUMLXX = SUMLXX + (LOG(Z(J, 1)) ^ 2): REM SIGMA LOG(X^2)
2620 SUMY = SUMY + Z(J, 2): REM SIGMA Y
2625 SUMLYY = SUMLYY + (LOG(Z(J, 2)) ^ 2): REM SIGMA LOG(Y^2)
2630 NEXT J
2640 M(1) = SUMX / NROW
2650 M(2) = (SUMXX - SUMX ^ 2 / NROW) / (NROW - 1)
2660 M(3) = SUMLY / NROW
2670 M(4) = (SUMYY - SUMLY ^ 2 / NROW) / (NROW - 1)
2680 M(6) = SUMLX / NROW
2690 ON NTYP GOTO 2700, 2750
2700 FACTB = (SUMXLY - SUMX * SUMLY / NROW) / (SUMXX - SUMX ^ 2 /
   NROW)
2710 FACTA = EXP(M(3) - FACTB * M(1) )
2720 A = INT(FACTA * 1000 + .2) / 1000: B INT(FACTB * 1000 + .2) /
   1000
2730 LOCATE 22, 30: PRINT "EQN. Y = "; A; "* e ^ ("; B; " * X)"
2740 GOTO 2790
2750 FACTB = (SUMLXY - SUMLX * SUMLY / NROW) / (SUMLXX - SUMLX ^ 2 /
   NROW
2760 FACTA = EXP(M(3) - FACTB * M(6) )
2770 A = INT(FACTA * 1000 + .2) / 1000: B = INT(FACTB * 1000 + .2) /
   1000
2780 LOCATE 22, 30: PRINT "EQN. Y = "; A; "* X ^ ("; B;")"
2790 FOR P = XMIN TO XMAX STEP (XMAX - XMIN) / 100: IF P = 0 THEN
   2870
2800 X1 = (P - XMIN) * 50 / XTC + X0: IF X1 > 570 THEN 2870
2810 IF NTYP = 2 THEN 2830
```

```
2820 Y1 = (FACTA * EXP(FACTB * P)) * 12 / YTC + Y0: GOTO 2840
2830 Y1 = (FACTA * P ^ FACTB) * 12 / YTC + Y0
2840 IF Y1 > 170 THEN 2870
2850 IF SCR% = 2 THEN RGB% = 12 ELSE RGB% = 1
2860 PSET (X1, Y1), RGB%
2870 NEXT P
2880 XBAR = SUMX / NROW
2885 STDX = SQR((SUMXX - NROW * XBAR ^ 2) / (NROW - 1))
2890 YBAR = SUMY / NROW
2895 STDY = SQR((SUMYY - NROW * YBAR ^ 2) / (NROW - 1))
2900 IF NTYP = 2 THEN 2940
2910 RT = (SUMXLY - SUMX * SUMLY / NROW)
2920 RB = SQR((SUMXX - SUMX ^ 2 / NROW) * (SUMLYY - SUMLY ^ 2 /
   NROW))
2930 GOTO 2960
2940 RT = (SUMLXY - SUMLX * SUMLY / NROW)
2950 RB = SQR((SUMLXX - SUMLX ^ 2 / NROW) * (SUMLYY - SUMLY ^ 2 /
   NROW))
2960 RXY = RT / RB
2970 CLOSE : OPEN "O", #3, DDSK$ + DFILEN$ + ".LIN"
2980 PRINT #3, NTYP + 1
2990 PRINT #3, FACTA
3000 PRINT #3, FACTB
3010 CLOSE
4000 A$ = INPUT$(1)
4010 IF A$ = " " THEN 4000 ELSE 4020
4020 LOCATE 24, 10: PRINT "Do you wish Printout of Data (Y/N)";
4030 A$ = INPUT$(1)
4040 IF INSTR("YNyn", A$) = 0 THEN 4030
4050 LOCATE 24, 10: PRINT " ";
4060 IF A$ = "Y" OR A$ = "y" THEN 6500
4070 GOTO 9530
```

"GRAPH-B3"—computer printout for exponential curve 'EXPCUR01'

```
Curve Fitting
DATA File : EXPCUR01
Exponential Curve Fit
****************************************
Factor A = 2
Factor B = 1.5
Mean X = .55
Mean Y = 4.999101
Standard Deviation X = .302765
Standard Deviation Y = 2.226885
Coefficient of Correlation = 1
Determination Coefficient = 1.000001
```

X-Data	Y-Data	Y-Calc	Residual
0.100	2.324	2.324	0.000
0.200	2.700	2.700	0.000
0.300	3.137	3.137	0.000
0.400	3.644	3.644	• 0.000
0.500	4.234	4.234	• 0.000
0.600	4.919	4.919	• 0.000
0.700	5.715	5.715	• 0.000
0.800	6.640	6.640	• 0.000
0.900	7.715	7.714	0.001
1.000	8.963	8.963	0.000

"GRAPH-B3"—computer printout for power curve 'POWERC01'

```
Curve Fitting
DATA File : POWERC01
Power Curve Fit
*******************************************
Factor A = 2
Factor B = 2
Mean X = 5.5
Mean Y = 77
Standard Deviation X = 3.02765
Standard Deviation Y = 68.34715
Coefficient of Correlation = .9999997
Determination Coefficient = .9999994
```

X-Data	Y-Data	Y-Calc	Residual
1.000	2.000	2.000	• 0.000
2.000	8.000	8.000	• 0.000
3.000	18.000	18.000	• 0.000
4.000	32.000	32.000	• 0.000
5.000	50.000	50.000	• 0.000
6.000	72.000	72.000	0.000
7.000	98.000	98.000	0.000
8.000	128.000	128.000	0.000
9.000	162.000	162.000	0.000
10.000	200.000	200.000	0.000

APPENDIX B.4:
CUBIC SPLINE COMPUTER PROGRAM "GRAPH-B4"

This program was discussed in Section 2.2.3 and fits a smoothed cubic spline through each of the data points in one data set.

Cubic spline

(a) Initialization and control Line numbers 10–90

See notes for section (a) of Appendix B.1. Since the fitting of spline curves to data requires matrix algebra, several arrays are also initialized in this section. Note that due to memory considerations, 20 is the maximum number of data points or rows.

(b) Data loading Line numbers 100–180

See notes for section (b) of Appendix B.1.

(c) Data check Line numbers 200–440

See notes for section (c) of Appendix B.1

(d) Text file entry Line numbers 500–580

See notes for section (d) of Appendix B.1

(e) Graph presentation selection Line numbers 600–750

See notes for section (e) of Appendix B.1

(f) Line graph display Line numbers 1000–1380

See notes for section (f) of Appendix B.1.

(g) Cubic spline curve Line numbers 3000–3940

This segment of code is described in Section 2.2.3 and uses Eqts 2.13– 2.15. The initial cubic spline polygon vertices and conditions are specified and then a curve fitted through the data points. The data points and the interpolated data curve points are saved in a data file with a **'.CUB'** extension for future plotting.

(h) Data printout? Line numbers 4000–4090

At the completion of the plot, the user can obtain a hardcopy printout by pressing the [Shift] and [Prt sc] keys simultaneously. The user also has an opportunity to obtain a printout of the data at Line 6500.

(i) Printout of data Line numbers 6500–6670

Please note this segment of code only uses Lines 6500–6570 of Appendix A.7. The remainder are new lines specific to this program.

This program also requires the routines (k) to (z) of Appendix B.2.

"GRAPH-B4"—BASIC program

```
10 REM <GRAPH-B4> Presentation Graphics Program B.4 - Line Graphs
14 REM (c) Copyright P.H.Milne 1990 **** - Cubic Spline ****
16 REM ALL RIGHTS RESERVED **** - 1 DATA Set Only ****
20 REM VERSION PC-1.00, 1990 : *** Maximum 20 Rows of Data ***
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
```

```
70 X0 = 70: Y0 = 50: A$ = "*": DATBX = 0: J = 1
80 DIM Z(20, 6), R(3, 100), J(100), CL%(5)
85 DIM N(20, 6), B(3, 20), U(3, 20), L(20), F(4, 2)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Cubic Spline"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
135 IF ND > 1 THEN 9590
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 150
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10 40: PRINT ": Max = "; XMAX
240 LOCATE 12 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12 40: PRINT ": Max = "; YMAX
260 LOCATE 15 10: PRINT "Change Default Values (Y/N) ";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 260
290 IF A$ = "N" OR A$ = "n" THEN 340
300 LOCATE 17, 10: INPUT "New X-AXIS (X-min, X-max) "; XMIN, XMAX
310 IF XMIN >= XMAX THEN 300
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "; YMIN, YMAX
330 IF YMIN >= YMAX THEN 320
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 500
370 YTC = (YMAX - YMIN) / 10
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 LOCATE 12, 10:
541 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
544 IF YLAB$ = "v" THEN YLAB$ = "V"
```

```
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS : GOSUB 9000
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 8, 10: PRINT "<1> Line Chart with Lines and Markers"
630 LOCATE 10, 10: PRINT "<2> Lines only"
640 LOCATE 12, 10: PRINT "<3> Markers only"
650 LOCATE 14, 10: PRINT "<4> Lines and Markers with Horizontal
   Grid"
660 LOCATE 16, 10:
665 PRINT "<5> Lines and Markers with Horizontal & Vertical Grid"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12345", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740, 750
710 GLIN = 1: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
720 GLIN = 1: MARK = 0: HGRID = 0: VGRID = 0: GOTO 800
730 GLIN = 0: MARK = 1: HGRID = 0: VGRID = 0: GOTO 800
740 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 800
750 GLIN = 1: MARK = 1: HGRID = 1: VGRID = 1: YLAB$ = "V": GOTO 800
800 CLS : GOSUB 9000
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 J = 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = YO - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1060 REM
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 1180
1110 J = 1
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 REM
1180 IF GLIN = 0 THEN 3000
1200 J = 1
1210 FOR K = 1 TO NROW - 1
1220 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1230 Y1 = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
```

```
1240 X2 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0
1250 Y2 = (Z(K + 1, J + 1) - YMIN) * 12 / YTC + Y0
1260 ON J GOTO 1270, 1280, 1290, 1300, 1310
1270 ST = &HFFFF: CL% = 4: GOTO 1320
1280 ST = &HCCCC: CL% = 1: GOTO 1320
1290 ST = &HF0FF: CL% = 2: GOTO 1320
1300 ST = &HCCFF: CL% = 3: GOTO 1320
1310 ST = &HCCFC: CL% = 5
1320 IF X1 > 560 THEN 1360
1330 IF Y1 > 160 THEN 1360
1340 PSET (X1, Y1)
1350 LINE - (X2,
1370 REM

3000 NPOLY = NROW: REM No. of Polygon Vertices
3010 SDIMN = 2: REM No. of Space Dimensions
3020 ZCUBS = 5: REM No. of Points on Cubic Spline
3030 FUNC1 = 1: REM Relaxed Initial End Condition
3040 FUNC2 = 1: REM Relaxed Final End Condition
3050 IF FUNC1 = 1 THEN 3090
3060 N(1, 2) = 1: N(1, 3) = 0
3070 FOR K = 1 TO SDIMN: B(K, 1) = U(K, 1): NEXT K
3080 GOTO 3100
3090 N(1, 2) = 1: N(1, 3) = .5
3100 FOR J = 1 TO NPOLY - 1
3110 L(J) = SQR((Z(J + 1, 1) - Z(J, 1)) ^ 2 + (Z(J + 1, 2) - Z(J,
   2)) ^ 2)
3120 NEXT J
3130 IF FUNC1 = 2 THEN 3170
3140 FOR K = 1 TO SDIMN
3150 B(K, 1) = (3 / (2 * L(l))) * (Z(2, K) - Z(1, K))
3160 NEXT K
3170 IF FUNC2 = 1 THEN 3230
3180 N(NPOLY, 1) = 0: N(NPOLY, 2) = 0
3190 FOR K = 1 TO SDIMN
3200 B(K, NPOLY) = U(K, NPOLY)
3210 NEXT K
3220 GOTO 3270
3230 N(NPOLY, 1) = 2: N(NPOLY, 2) = 4
3240 FOR K = 1 TO SDIMN
3250 B(K, NPOLY) = (6 / L(NPOLY - 1)) * (Z(NPOLY, K) - Z(NPOLY - 1,
   K))
3260 NEXT K
3270 FOR J = 2 TO NPOLY - 1
3280 N(J, 1) = L(J)
3290 N(J, 2) = 2 * (L(J) + L(J - 1))
3300 N(J, 3) = L(J - 1)
3310 FOR K = 1 TO SDIMN
3320 B(K, J) = L(J - 1) ^ 2 * (Z(J + 1, K) - Z(J, K))
```

```
3330 B(K, J) = B(K, J) + L(J) ^ 2 * (Z(J, K) - Z(J - 1, K))
3340 B(K, J) = 3 * B(K, J) / (L(J) * L(J - 1))
3350 NEXT K
3360 NEXT J
3370 FOR I = 2 TO NPOLY
3380 IF N(I, 1) = 0 THEN 3490
3390 D = N(I - 1, 2) / N(I, 1)
3400 FOR K = 1 TO 3
3410 N(I, K) = N(I, K) * D - N(I - 1, K + 1)
3420 B(K, I) = B(K, I) * D - B(K, I - 1)
3430 NEXT K
3440 Q = N(I, 2)
3450 FOR K = 1 TO 3
3460 N(I, K) = N(I, K) / Q
3470 B(K, I) = B(K, I) / Q
3480 NEXT K
3490 NEXT I
3500 FOR K = 1 TO SDIMN
3510 FOR J = 0 TO NPOLY - 1
3511 U1 = U(K, NPOLY + 1 - J)
3512 B1 = B(K, NPOLY - J)
3520 U(K, NPOLY - J) = (B1 - N(NPOLY - J, 3) * U1) / N(NPOLY - J, 2)
3530 NEXT J
3540 NEXT K
3550 I = 1
3560 CLOSE: OPEN "O", #3, DDSK$ + DFILEN$ + ".CUB"
3570 FOR J = 1 TO NPOLY - 1
3580 FOR K = 1 TO SDIMN
3590 F(1, K) = Z(J, K)
3600 F(2, K) = U(K, J)
3610 F(3, K) = (3 / L(J)) ^ 2) * (Z(J + 1, K) - Z(J, K))
3620 F(3, K) = F(3, K) - (1 / L(J)) * (U(K, J + 1) + 2 * U(K, J))
3630 F(4, K) = (-2 / L(J) ^ 3) * (Z(J + 1, K) - Z(J, K))
3640 F(4, K) = F(4, K) + (1 / L(J) ^ 2) * (U(K, J + 1) + U(K, J))
3650 NEXT K
3660 FOR T = 0 TO L(J) STEP L(J) / ZCUBS
3700 FOR K = 1 TO SDIMN
3711 R1 = F(1, K) + F(2, K) * T
3710 R(K, I) = R1 + F(3, K) * (T ^ 2) + F(4, K) * (T ^ 3)
3720 R(K, I) = INT(Z(NPOLY, K) * 1000 + .1) / 1000
3730 NEXT K
3740 IF T = 0 AND R(1, I) =R(1, I - 1) THEN 3770
3750 PRINT #3, R(1, I), R(2, I)
3760 I = I + 1
3770 NEXT T
3780 NEXT J
3800 FOR K = 1 TO SDIMN
3810 R(K, I) = INT(Z(NPOLY, K) * 1000 + .1) / 1000
3820 NEXT K
```

```
3830 IF R(1, I) = R(1 / I - 1) THEN 3850
3840 PRINT #3, R(1, I), R(2, I)
3850 CLOSE
3860 FOR J2 = 1 TO I - 1
3870 X1 = R(1, J2) * 50 / XTC + X0
3880 Y1 = R(2, J2) * 12 / YTC + Y0
3890 X2 = R(1, J2 + 1) * 50 / XTC + X0
3900 Y2 = R(2, J2 + 1) * 12 / YTC + Y0
3910 PSET (X1, Y1)
3920 LINE - (X2, Y2)
3930 NEXT J2
4000 A$ = INPUT$(1)
4010 IF A$ = "" THEN 4000 ELSE 4020
4020 LOCATE 24, 10: PRINT "Do you wish Printout of Data (Y/N) ";
4030 A$ = INPUT$(1)
4040 IF INSTR("YNyn", A$) = 0 THEN 4030
4050 IF A$ = "Y" OR A$ = "y" THEN 6500
4060 LOCATE 24, 10: PRINT " ";
4070 GOTO 9530

6500 LOCATE 24, 10: PRINT " ";
6510 LOCATE 24, 10: PRINT "Switch Printer Online - Continue (Y/N)";
6520 P$ = INPUT$(1)
6530 IF INSTR("YNyn", P$) = 0 THEN 6500
6540 IF P$ = "N" OR P$ = "n" THEN 9530
6550 LPRINT PC$
6560 LPRINT "DATA File : "; DFILEN$: LPRINT TITLE$
6570 LPRINT "*****************************************"
6580 LPRINT TAB(5); "X-data"; TAB(15); "Y-data"
6590 LPRINT "========================================"
6600 FOR K = 1 TO I
6610 IF R(1, K) = R(1, K - 1) THEN 6630
6620 LPRINT USING "######.###"; R(1, K); R(2, K)
6630 NEXT K
6640 LPRINT "======================================="
6650 LPRINT
6660 LOCATE 24, 10: PRINT " ";
6670 GOTO 9530
```

"GRAPH-B4"—computer printout for cubic spline 'CUBICS03'

```
Cubic Spline
DATA File : CUBICS03
Cubic Spline Test
*********************************************
```

X-data	Y-data
10.000	10.000

X-data	Y-data
11.973	37.739
13.953	62.544
15.947	81.479
17.960	91.609
20.000	90.000
21.943	76.610
23.915	56.355
25.915	35.295
27.944	19.490
30.000	15.000
31.946	24.489
33.918	42.692
35.916	63.152
37.943	79.407
40.000	85.000
41.940	76.866
43.909	60.109
45.908	40.920
47.938	25.487
50.000	20.000
51.937	27.375
53.904	42.808
55.902	60.553
57.934	74.865
60.000	80.000
61.930	73.309
63.893	59.240
65.891	43.015
67.926	29.860
70.000	25.000
71.927	30.799
73.890	43.272
75.890	57.846
77.926	69.946
80.000	75.000
81.909	70.845
83.856	60.730
85.849	48.192
87.895	36.770
90.000	30.000
91.936	30.137
93.915	35.516
95.925	44.827

X-data	Y-data
97.957	56.758
100.000	70.000

COMPUTER PROGRAM "GPLOT-B4"

To convert the program **"GRAPH-B4"** from screen graphics to an **HPGL** plotter the following changes are required:

replace segment (f) with the code for Lines 1000–1370, which are given at the end of this section
replace segment (g) with the code for Lines 3000–3940, which are given at the end of this section
replace segment (m) with Appendix A.28
replace segment (n) with Appendix A.29
replace segment (o) with Appendix A.30
alter Lines 8270–8290 in segment (q) as shown in the code at the end of this section
replace segment (y) with Appendix A.32
delete segments (h), (i), (t) and (z)
add segments (aa)–(dd) from **"GPLOT-B1"**

"GPLOT-B4"—BASIC program—segments (f), (g) and part (q) only

```
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1005 GOSUB 50500: REM Header
1010 GOSUB 7500: REM Titles
1020 J = 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1045 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY - 2; ";": GOSUB 60000
1050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1060 REM
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 IF MARK = 0 THEN 1180
1110 J = 1
1120 FOR K = 1 TO NROW
1130 AVGX = (Z(K, 1) - XMIN) * 50 / XTC + X0
1135 IF AVGX > 570 THEN 1160
1140 AVGY = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1145 IF AVGY > 170 THEN 1160
1150 GOSUB 40000
1160 NEXT K
1170 REM
1180 IF GLIN = 0 THEN 3000
1200 J = 1
1210 FOR K = 1 TO NROW - 1
1220 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1230 Y1 = (Z(K, J + 1) - YMIN) * 12 / YTC + Y0
1240 X2 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0
```

```
1250 Y2 = (Z(K + 1, J + 1) - YMIN) * 12 / YTC + Y0
1260 ON J GOTO 1270, 1280, 1290, 1300, 1310
1270 ST = &HFFFF: CL% = 4: GOTO 1320
1280 ST = &HCCCC: CL% = 1: GOTO 1320
1290 ST = &HF0FF: CL% = 2: GOTO 1320
1300 ST = &HCCFF: CL% = 3: GOTO 1320
1310 ST = &HCCFC: CL% = 5
1320 IF X1 > 560 THEN 1360
1330 IF Y1 > 160 THEN 1360
1340 PRINT #3, "PU; SP1;": GOSUB 60000
1350 GOSUB 51030
1360 NEXT K
1370 REM
3000 REM ***** Cubic Spline *****
3550 I = 1
3560 OPEN "I", #2, DDSK$ + DFILEN$ + ".CUB"
3750 INPUT #2, R(1, I), R(2, I)
3760 I = I + 1
3770 GOTO 3750
3850 CLOSE #2
3860 FOR J2 = 1 TO I - 2
3870 X1 = R(1, J2) * 50 / XTC + X0
3880 Y1 = R(2, J2) * 12 / YTC + Y0
3890 X2 = R(1, J2 + 1) * 50 / XTC + X0
3900 Y2 = R(2, J2 + 1) * 12 / YTC + Y0
3910 PRINT #3, "PU;SP2;"
3920 GOSUB 51030
3930 NEXT J2
3935 PRINT #3, "PU;PA0,0;SP0;": CLOSE
3940 GOTO 9530

8270 IF INSTR("YNyn", A$) = 0 THEN 8260
8275 IF A$ = "N" OR A$ = "n" THEN 8800
8280 OPEN "I", #2, DDSK$ + DFILEN$ + ".CUB"
8284 CLOSE #2: GOTO 8290
8285 LOCATE 10, 10: PRINT "You must prepare Cubic Spline Data first"
8286 GOTO 9530
8290 LOCATE 10, 10: PRINT "Please wait … reading DATA"
```

APPENDIX B.5:

AREA CHARTS COMPUTER PROGRAM “GRAPH-B5”

This program was described in Section 2.3 and gives the user the option to plot three different types of area charts.

(a) Initialization and control

Line numbers 10–90

See notes for section (a) of Appendix B.1. Note two additional arrays are required to store data for the two stacked area charts.

(b) Data loading Line numbers 100–180

See notes for section (b) of Appendix B.1.

(c) Data check Line numbers 200–440

See notes for section (c) of Appendix B.1.

(d) Text file entry Line numbers 500–580

See notes for section (d) of Appendix B.1.

(e) Graph presentation selection Line numbers 600–750

See general notes for section (e) of Appendix B.1. The user chooses in this segment how to display the data: as an area chart, a stacked area chart or a 100% stacked area chart.

(f) Stacked area set up Line numbers 800–990

These two segments of code calculate the maximum scale value of *Y* for the display **(YSCM),** depending on the type of stacked area chart.

(g) Graph display Line numbers 1000–1100

See the early notes of section (f) of Appendix B.1 for setting up the display.

(h) Area chart display Line numbers 1500–1910

This segment of code was described in Section 2.3.1. The areas under the data set values will be shaded either in colour or by hatching depending on the type of screen used.

(i) Stacked area chart Line numbers 2000–2200

This segment of code was described in Section 2.3.2 for stacked area charts and Section 2.3.3 for 100% stacked area charts. The areas will be shaded as described in section (h) above.

This program also requires the routines (k) to (z) listed in Appendix B.2.

"GRAPH-B5"—BASIC program

```
10 REM <GRAPH-B5> Presentation Graphics Program B.5 - Area Charts
14 REM (C) Copyright P.H.Milne 1990
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: *** Maximum 100 Rows of Data ***
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
```

```
60 GOSUB 8000: REM Check Screen & Disc
70 X0 = 70: Y0 = 50: A$ = "*": DATBX = 1
80 DIM Z(100, 6), ZD(100, 6), ZP(100, 6), CL%(5)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Area Chart"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N)";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 150
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS: Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N)";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 260
290 IF A$ = "N" OR A$ = "n" THEN 340
300 LOCATE 17, 10: INPUT "New X-AXIS (X-min, X-max) "
310 IF XMIN >= XMAX THEN 300
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "
330 IF YMIN >= YMAX THEN 320
340 IF XMIN > 0 THEN XMIN = 0
345 IF XMIN < 0 THEN XMIN = XMIN - 10
350 IF YMIN > 0 THEN YMIN = 0
355 IF YMIN < 0 THEN YMIN = YMIN - 10
360 XTC = (XMAX - XMIN) / 10: TCC = XTC: GOSUB 7100
370 XTC = TCC: XM = 500
380 YTC = (YMAX - YMIN) / 10: TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) " LYAXI$
540 LOCATE 10, 12:
541 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
```

```
544 IF YLAB$ = "v" THEN YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA"; I; "(Max 9)";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS : GOSUB 9000
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 8, 10: PRINT "<1> Area Chart"
630 LOCATE 10, 10: PRINT "<2> Area Chart with Horizontal Grid"
640 LOCATE 12, 10: PRINT "<3> Stacked Area Chart"
650 LOCATE 14, 10: PRINT "<4> Stacked Area Chart with Horizontal
   Grid"
660 LOCATE 16, 10: PRINT "<5> 100% Stacked Area Chart"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12345", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740, 750
710 GLIN = 0: MARK = 0: HGRID = 0: VGRID = 0: GOTO 1000
720 GLIN = 0: MARK = 0: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 1000
730 GLIN = 0: MARK = 0: HGRID = 0: VGRID = 0: GOTO 800
740 GLIN = 0: MARK = 0: HGRID = 1: VGRID = 0: YLAB$ = "V": GOTO 800
750 GLIN = 0: MARK = 0: HGRID = 0: VGRID = 0: GOTO 900
799 REM *** Setup Stacked Area Chart ***
800 FOR I = 1 TO NROW: YSCM = 0
810 FOR K = 2 TO NCOL
820 YSCM = YSCM + Z(I, K)
830 IF YSCM > YSCMAX THEN YSCMAX = YSCM
840 NEXT K
850 NEXT I
860 YTC = YSCMAX / 10
870 TCC = YTC: GOSUB 7100
880 YTC = TCC: YM = 120
890 GOTO 1000
899 REM *** Setup 100% Stacked Area Chart ***
900 FOR I = 1 TO NROW: YSCM = 0
910 FOR K = 2 TO NCOL
920 YSCM = YSCM + Z(I, K)
930 ZP(I, K) = YSCM
940 NEXT K
950 FOR J = 2 TO NCOL
960 ZD(I, J) = ZP(I, J) / YSCM
970 NEXT J
980 NEXT I
990 YTC = 10
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 FOR J = 1 TO NCOL - 1
```

```
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1060 NEXT J
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 ON NSEL GOTO 1500, 1500, 2000, 2000, 2000
1499 REM ***** Area Graph *****
1500 FOR J = NCOL TO 2 STEP -1
1510 X1 = (Z(1, 1) - XMIN) * 50 / XTC + X0: X2 = X1
1520 Y1 = Y0 + 1
1530 Y2 = (Z(1, J) - YMIN) * 12 / YTC + Y0: YPT = Y2
1540 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1550 FOR K = 1 TO NROW - 1
1560 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1570 Y1 = (Z(K, J) - YMIN) * 12 / YTC + Y0
1580 X2 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0
1590 Y2 = (Z(K + 1, J) - YMIN) * 12 / YTC + Y0
1600 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1610 NEXT K
1620 X1 = X2: Y1 = Y2
1630 IF SCR% <> 2 THEN 1710
1640 X2 = X1: Y2 = Y0 + 1
1650 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1660 X1 = X2: Y1 = Y2
1670 X2 = (Z(1, 1) - XMIN) * 50 / XTC + X0
1680 Y2 = Y1
1690 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1700 PAINT (X2 + 5, Y2 + 5), CL%(J - 1): GOTO 1880
1710 X2 = X1: IF J = 2 THEN 1810
1720 Y2 = (Z(NROW, J - 1) - YMIN) * 12 / YTC - Y0
1730 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1740 FOR K = NROW TO 2 STEP -1
1750 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1760 Y1 = (Z(K, J - 1) - YMIN) * 12 / YTC + Y0
1770 X2 = (Z(K - 1, 1) - XMIN) * 50 / XTC + X0
1780 Y2 = (Z(K - 1, J - l) - YMIN) * 12 / YTC + Y0
1790 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1800 NEXT K: GOTO 1870
1810 Y2 = Y0 + 1
1820 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1830 X1 = X2: Y1 = Y2
1840 X2 = (Z(1, 1) - XMIN) * 50 / XTC + X0
1850 Y2 = Y1
1860 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
1870 PAINT (X2 + 3, Y2 + 3), TILE$(J - 1)
1880 NEXT J
1890 IF SCR% = 2 THEN RGB% = 15 ELSE RGB% = 1
1900 LINE (X0, Y0 + YM) - (X0, Y0), RGB%
```

```
1910 GOTO 9500
1999 REM ***** Stacked Area Chart *****
2000 FOR K = 1 TO NROW - 1
2010 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0: X2 = X1
2020 Y1 = Y0 + 1: YY1 = Y1
2030 FOR J = 2 TO NCOL
2040 XX1 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0: XX2 = XX1
2050 IF NSEL = 5 THEN 2080
2060 Y2 = Y1 + (Z(K, J) - YMIN) * 12 / YTC
2070 YY2 = YY1 + (Z(K + 1, J) - YMIN) * 12 / YTC: GOTO 2100
2080 Y2 = Y0 + (ZD(K, J) - YMIN) * 120
2090 YY2 = Y0 + (ZD(K + 1, J) - YMIN) * 120
2100 LINE (X1, Y1) - (X2, Y2), CL%(J - 1)
2110 LINE - (XX2, YY2), CL%(J - 1)
2120 LINE - (XX1, YY1), CL%(J - 1)
2130 LINE - (X1, Y1), CL%(J - 1)
2140 IF SCR% <> 2 THEN 2160
2150 PAINT (XX1 - 2, (YY1 + YY2) / 2), CL%(J - 1): GOTO 2170
2160 PAINT (X1 + 1, (YY1 + YY2) / 2), TILE$(J - 1)
2170 Y1 = Y2: YY1 = YY2
2180 NEXT J
2190 NEXT K
2200 GOTO 9500
```

COMPUTER PROGRAM "GPLOT-B5"

To convert the program **"GRAPH-B5"** from screen graphics to an HPGL plotter the following changes are required:

replace segment (g) with the code for Lines 1000–1100, which are given at the end of this section
replace segment (h) with the code for Lines 1500–1910, which are given at the end of this section
replace segment (i) with the code for Lines 2000–2200, which are given at the end of this section
replace segment (m) with Appendix A.28
replace segment (n) with Appendix A.29
replace segment (o) with Appendix A.30
replace segment (y) with Appendix A.32
delete segments (t) and (z)
add segments (aa)–(dd) from **"GPLOT-B1"**
add Appendix A.22

Note that this Appendix uses a **PM** command which may not work with filling irregular polygons on your specific HPGL plotter.

"GPLOT-B5"—BASIC program—segments (g), (h) and (i) only

```
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1005 GOSUB 50500: REM Header
1010 GOSUB 7500: REM Titles
1020 FOR J = 1 TO NCOL - 1
```

```
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1045 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY - 2; ";": GOSUB 60000
1050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1060 NEXT J
1070 IF HGRID = 0 THEN 1100
1080 GOSUB 7600: REM Draw Grid
1100 ON NSEL GOTO 1500, 1500, 2000, 2000, 2000
1499 REM ***** Area Graph *****
1500 FOR J = NCOL TO 2 STEP -1
1510 X1 = (Z(1, 1) - XMIN) * 50 / XTC + X0
1520 Y1 = (Z(1, J) - YMIN) * 12 / YTC + Y0
1525 PRINT #3, "PU; PA"; X1; ","; Y1; "; PM0; OE"
1550 FOR K = 1 TO NROW - 1
1560 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1570 Y1 = (Z(K, J) - YMIN) * 12 / YTC + Y0
1580 X2 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0
1590 Y2 = (Z(K + 1, J) - YMIN) * 12 / YTC + Y0
1600 GOSUB 51000
1610 NEXT K
1620 X1 = X2: Y1 = Y2
1710 X2 = X1: IF J = 2 THEN 1810
1720 Y2 = (Z(NROW, J - 1) - YMIN) * 12 / YTC + Y0
1730 GOSUB 51000
1740 FOR K = NROW TO 2 STEP -1
1750 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0
1760 Y1 = (Z(K, J - 1) - YMIN) * 12 / YTC + Y0
1770 X2 = (Z(K - 1, 1) - XMIN) * 50 / XTC + X0
1780 Y2 = (Z(K - 1, J - 1) - YMIN) * 12 / YTC + Y0
1790 GOSUB 51000
1800 NEXT K: GOTO 1870
1810 Y2 = Y0
1820 GOSUB 51000
1830 X1 = X2: Y1 = Y2
1840 X2 = (Z(1, 1) - XMIN) * 50 / XTC + X0
1850 Y2 = Y1
1860 GOSUB 51000
1870 GOSUB 52030
1880 NEXT J
1900 PRINT #3, "PU;PA0,0;SP0;": CLOSE #3
1910 GOTO 9530
1999 REM ***** Stacked Area Chart *****
2000 FOR K = 1 TO NROW - 1
2010 X1 = (Z(K, 1) - XMIN) * 50 / XTC + X0: X2 = X1
2020 Y1 = Y0: YY1 = Y1
2025 PRINT #3, "PU;PA"; X1; ","; Y1; ";PM0;OE;"
2030 FOR J = 2 TO NCOL
2040 XX1 = (Z(K + 1, 1) - XMIN) * 50 / XTC + X0: XX2 = XX1
2050 IF NSEL = 5 THEN 2080
```

```
2060 Y2 = Y1 + (Z(K, J) - YMIN) * 12 / YTC
2070 YY2 = YY1 + (Z(K + 1, J) - YMIN) * 12 / YTC: GOTO 2100
2080 Y2 = Y0 + (ZD(K, J) - YMIN) * 120
2090 YY2 = Y0 + (ZD(K + 1, J) - YMIN) * 120
2100 GOSUB 51000
2110 PRINT #3, "PA"; XX2; ","; YY2; ";"
2120 PRINT #3, "PA"; XX1; ","; YY1; ";"
2130 PRINT #3, "PA"; X1; ","; Y1; ";"
2160 GOSUB 52030
2170 Y1 = Y2: YY1 = YY2
2180 NEXT J
2190 NEXT K
2195 PRINT #3, "PU;PA0,0;SP0;": CLOSE #3
2200 GOTO 9530
```

Appendix C
Bar and column charts

APPENDIX C.1:
BAR AN D COLUMN CHARTS COMPUTER PROGRAM “GRAPH-C1”

This program was discussed in Chapter 3 and plots bar and column charts and their multiple and stacked variations.

Bar and column charts

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables that require to be initialized for use in the program. Three data arrays are required (**Z, ZD** and **ZP**) to cater for stacked data sets. Note that for clarity, no more than 5 data sets and fifteen rows of data should be used.

(b) Data loading Line numbers 100–180

See the notes on section (b) of Appendix B.1.

(c) Data check Line numbers 200–440

See the notes on section (c) of Appendix B.1.

(d) Text file entry Line numbers 500–585

This routine is only used if no text file is found in the previous section (c), or if the user wishes to change the title and graph axes labels.

(e) Column or bar chart? Line numbers 590–595

The user now has to choose which type of chart to display. If a column chart is chosen, the program continues at Line 600, and if a bar chart continues at Line 2500.

(f) Column chart presentation? Line numbers 600–740

The user now has to choose how to display the column chart data, either as a single or multiple column chart, or as a stacked or 100% stacked column chart. The user can also display a background grid if required for scaling. Once these selections have been made, the program branches at Line 740, depending on the user choice of presentation.

(g) Data preparation for columns Line numbers 750–990

This segment of code contains three sets of routines to prepare data for a single column chart or for either of the stacked column charts.

(h) Column chart display Line numbers 1000–1670

This is the column chart part of the program and uses several of the graphics presentation routines described in Chapter 1, to set up the screen **(GOSUB 7300),** title the graph **(GOSUB 7500),** and draw a background grid if required **(GOSUB 7600)**. Colours and monochrome hatching patterns are then chosen for the display. If a single or multiple column chart display was chosen, this is drawn in Lines 1200–1410. If a stacked column chart was chosen, then this is drawn in Lines 1500–1670.

(i) Bar chart presentation? Line numbers 2500–2740

The user now has to choose how to display the bar chart, either as a single or multiple bar chart, or as a stacked or 100% stacked bar chart. The user can also display a vertical grid if required for scaling.

(j) Data preparation for bars Line numbers 2750–2990

This segment of code contains three sets of routines to prepare data for a single bar chart or for either of the stacked bar charts.

(k) Bar chart display Line numbers 3000–3670

This is the bar chart part of the program and uses several of the graphics presentation routines described in Chapter 1. To set up the screen for bar charts, the *X* and *Y* axes have to be reversed, and this is contained in a routine at Line 7800. The graph is titled using **(GOSUB 7500)** where some changes are required due to reversing the *X* and *Y* axes. A background grid is also drawn if required **(GOSUB 7600)**. Colours and monochrome hatching patterns are then chosen for the display. If a single or multiple bar chart display was chosen, this is drawn in Lines 3200–3410. If a stacked bar chart was chosen, this is drawn in Lines 3500–3670.

(l) Appendix A.8 Line numbers 7000–7090
(m) Appendix A.9 Line numbers 7100–7290
(n) Appendix A.10 Line numbers 7300–7490
(o) Appendix A.11 Line numbers 7500–7590
Note changes required to reverse *X*-axis and *Y*-axis labels.
(p) Appendix A.12 Line numbers 7600–7710

(q) Reverse *X* and *Y* axes Line numbers 7800–7990

This routine is similar to (n) above, except that the *X* and *Y* axes ranges are reversed, with the *X*-range now plotted vertically and the *Y*-range plotted horizontally.
The program also requires the following routines.

(r) Appendix A.2	Line numbers 8000–8180
(s) Appendix A.3	Line numbers 8200–8390
(t) Appendix A.4	Line numbers 8400–8490
(u) Appendix A.5	Line numbers 8500–8590
(v) Appendix A.14	Line numbers 8600–8660
(w) Appendix A.3	Line numbers 8800–8990
(x) Appendix A.15	Line numbers 9000–9050
(y) Appendix A.16	Line numbers 9500–9600
(z) Appendix A.17	Line numbers 10000–10310
(aa) Appendix A.18	Line numbers 40000–40590
(bb) Appendix A.19	Line numbers 55000–56960

“GRAPH-C1”—BASIC program

```
10 REM <GRAPH-C1> Presentation Graphics Program C.1
14 REM (C) Copyright P.H.Milne 1990 - Bar/Column Charts
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
70 X0 = 70: Y0 = 50: DATBX = 0: YSCM = 0: YSCMAX = 0
75 A$ = "*": DSET$ = "*": HIST$ = "N" : HGRID$ = "N": VGRID$ = "N"
80 DIM Z(15, 6), ZD(15, 6), ZP(15, 6), DAT$(5)
90 WINDOW (0, 0) - (639, 199)
100 CLS : PC$ = "Bar Chart"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 140
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% <> 2 THEN GOSUB 8600
200 CLS : GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N) ";
270 A$ = INPUT$(1)
280 IF INSTR("YNyn", A$) = 0 THEN 260
290 IF A$ = "N" OR A$ = "n" THEN 340
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-min, Y-max) "; YMIN, YMAX
```

```
330 IF YMIN >= YMAX THEN 320
350 IF YMIN > 0 THEN YMIN = 0
355 IF YMIN < 0 THEN YMIN = YMIN - 10
360 XTC = INT(500 / NROW): XND = INT(XTC / (ND + 1))
370 XM = 500
380 YTC = (YMAX - YMIN) / 10: TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 590
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 LOCATE 12, 10:
541 INPUT "Y-AXIS Title (H)oriz. or (V)ert. "; YLAB$
542 IF INSTR("HVhv", YLAB$) = 0 THEN 540
543 IF YLAB$ = "h" THEN YLAB$ = "H"
544 IF YLAB$ = "v" THEN YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 13 + I, 10:
561 PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
585 GOSUB 8400: REM Store Titles on Disc
590 LOCATE 22, 10: PRINT "Select Column Chart or Bar Chart (C/B)";
591 CH$ = INPUT$(1)
592 IF INSTR("CBcb", CH$) = 0 THEN 590
593 IF CH$ = "C" THEN CH$ = "C"
594 IF CH$ = "b" THEN CH$ = "B"
595 IF CH$ = "B" THEN 2500
600 CLS : PC$ = "Column Chart": GOSUB 9000
605 IF ND = 1 THEN NSEL = 1: GOTO 695
610 LOCATE 5, 10: PRINT "Select Column Chart Presentation :-"
620 LOCATE 8, 10: PRINT "<1> Single column chart"
630 LOCATE 10, 10: PRINT "<2> Multiple column chart"
640 LOCATE 12, 10: PRINT "<3> Stacked column chart"
650 LOCATE 14, 10: PRINT "<4> 100% stacked column chart"
670 LOCATE 16, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("1234", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
695 IF SCR% <> 2 THEN 740
700 LOCATE 20, 10: PRINT "Do you wish Horizontal Grid (Y/N)";
705 HGRID$ = INPUT$(1)
710 IF INSTR("YNyn", HGRID$) = 0 THEN 700
720 IF HGRID$ = "y" THEN HGRID$ = "Y"
```

```
730 IF HGRID$ = "n" THEN HGRID$ = "N"
732 IF HGRID$ = "Y" THEN YLAB$ = "V"
735 IF NSEL = 4 THEN YLAB$ = "V"
740 ON NSEL GOTO 750, 1000, 800, 900
749 REM *** Setup Single Column Chart/Histogram ***
750 IF ND = 1 THEN K = 2: GOTO 772
755 LOCATE 22, 10: PRINT "Which Data Set to plot 1-"; ND;
760 DSET$ = INPUT$(1)
765 IF INSTR("12345", DSET$) = 0 THEN 760
770 K = VAL(DSET$) + 1
772 LOCATE 24, 10: PRINT "Plot as Histogram (Y/N) ";
775 HIST$ = INPUT$(1)
780 IF INSTR("YNyn", HIST$) = 0 THEN 775
785 IF HIST$ = "y" THEN HIST$ = "Y"
790 IF HIST$ = "n" THEN HIST$ = "N"
795 GOTO 1000
799 REM *** Setup Stacked Column Chart ***
800 FOR I = 1 TO NROW: YSCM = 0
810 FOR K = 2 TO NCOL
820 YSCM = YSCM + (Z(I, K) - YMIN)
830 IF YSCM > YSCMAX THEN YSCMAX = YSCM
840 NEXT K
850 NEXT I
860 YTC = YSCMAX / 10
870 TCC = YTC: GOSUB 7100
880 YTC = TCC: YM = 120
890 GOTO 1000
899 REM *** Setup 100% Stacked Column Chart ***
900 FOR I = 1 TO NROW: YSCM = 0
910 FOR K = 2 TO NCOL
920 YSCM = YSCM + (Z(I, K) - YMIN)
930 ZP(I, K) = YSCM
940 NEXT K
950 FOR J = 2 TO NCOL
960 ZD(I, J) = ZP(I, J) / YSCM
970 NEXT J
980 NEXT I
990 YTC = 10
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: DATBX = 1: REM Titles
1015 IF NSEL = 1 THEN J = VAL(DSET$): GOTO 1030
1020 FOR J = 1 TO NCOL - 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1055 IF NSEL = 1 THEN 1070
1060 NEXT J
1070 IF HGRID$ = "N" THEN 1100
1080 GOSUB 7600: REM Draw Grid
```

```
1100 IF SCR% = 2 THEN CL(1) = 4 ELSE CL(1) = 1
1110 IF SCR% = 2 THEN CL(2) = 1 ELSE CL(2) = 1
1120 IF SCR% = 2 THEN CL(3) = 2 ELSE CL(3) = 1
1130 IF SCR% = 2 THEN CL(4) = 3 ELSE CL(4) = 1
1140 IF SCR% = 2 THEN CL(5) = 5 ELSE CL(5) = 1
1150 ON NSEL GOTO 1200, 1200, 1500, 1500
1200 FOR J = 1 TO NROW: IF NSEL = 1 THEN 1215
1210 FOR K = 2 TO NCOL
1215 IF NSEL = 1 THEN X1 = (J - 1) * XTC + X0 + 15
1220 IF NSEL = 2 THEN X1 = (J - 1) * XTC + (K - 1) * XND + X0
1225 IF HIST$ = "Y" THEN X1 = (J - 1) * XTC + X0 + 1
1230 Y1 = Y0 + 1
1240 IF HIST$ = "N" THEN X2 = X1 + (XND - 1)
1245 IF DSET$ <> "*" THEN X2 = X1 + XTC / 2
1250 IF HIST$ = "Y" THEN X2 = X1 + XTC
1255 Y2 = (Z(J, K) - YMIN) * 12 / YTC + Y0
1320 IF X1 > 560 THEN 1390
1330 IF Y1 > 160 THEN 1390
1340 IF SCR% <> 2 THEN 1360
1350 LINE (X1, Y1) - (X2, Y2), CL(K - 1), BF: GOTO 1380
1360 LINE (X1, Y1) - (X2, Y2), , B
1370 PAINT (X1 + 2, Y1 + 2), TILE$(K - 1)
1380 IF NSEL = 1 THEN 1400
1390 NEXT K
1400 NEXT J
1410 GOTO 9500
1499 REM ***** Stacked Column Charts *****
1500 FOR J = 1 TO NROW
1510 Y1 = Y0 + 1
1520 FOR K = 2 TO NCOL
1530 X1 = (J - 1) * XTC + X0 + 10
1540 Y1 = Y1
1550 X2 = J * XTC + X0
1560 IF NSEL = 3 THEN Y2 = Y1 + (Z(J, K) - YMIN) * 12 / YTC
1570 IF NSEL = 4 THEN Y2 = Y0 + ZD(J, K) * 120
1580 IF X1 > 560 THEN 1650
1590 IF Y1 > 160 THEN 1650
1600 IF SCR% <> 2 THEN 1620
1610 LINE (X1, Y1) - (X2, Y2), CL(K - 1), BF: GOTO 1640
1620 LINE (X1, Y1) - (X2, Y2), , B
1630 PAINT (X1 + 2, Y1 + 2), TILE$(K - 1)
1640 Y1 = Y2
1650 NEXT K
1660 NEXT J
1670 GOTO 9500
2499 REM ***** BAR CHART *****
2500 YTC = INT(120 / NROW)
2510 YND = INT(YTC / (ND + 1))
2520 YM = NROW * YTC
```

```
2530 XTC = (YMAX - YMIN) / 10
2340 TCC = XTC: GOSUB 7100
2550 XTC = TCC: XM = 500
2600 CLS: GOSUB 9000: IF ND = 1 THEN NSEL = 1: GOTO 2695
2610 LOCATE 5, 10: PRINT "Select Bar Chart Presentation :-"
2620 LOCATE 8, 10: PRINT "<1> Single bar chart"
2630 LOCATE 10, 10: PRINT "<2> Multiple bar chart"
2640 LOCATE 12, 10: PRINT "<3> Stacked bar chart"
2650 LOCATE 14, 10: PRINT "<4> 100% Stacked bar chart"
2670 LOCATE 16, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
2680 IF INSTR("1234", SEL$) = 0 THEN 2670
2690 NSEL = VAL(SEL$): HGRID$ = "N": VGRID$ = "N"
2695 IF SCR% <> 2 THEN 2740
2700 LOCATE 20, 10: PRINT "Do you wish Vertical Grid (Y/N) ";
2705 VGRID$ = INPUT$(1)
2710 IF INSTR("YNyn", VGRID$) = 0 THEN 2700
2720 IF VGRID$ = "y" THEN VGRID$ = "Y"
2730 IF VGRID$ = "n" THEN VGRID$ = "N"
2735 YLAB$ = "V"
2740 ON NSEL GOTO 2750, 3000, 2800, 2900
2749 REM *** Setup Single Bar Chart ***
2750 IF ND = 1 THEN K = 2: GOTO 3000
2760 LOCATE 22, 10: PRINT "Which Data Set to plot 1-"; ND;
2765 DSET$ = INPUT$(1)
2770 IF INSTR("12345", DSET$) = 0 THEN 2760
2780 K = VAL(DSET$) + 1
2790 GOTO 3000
2799 REM *** Setup Stacked Bar Chart ***
2800 FOR I = 1 TO NROW: XSCM = 0
2810 FOR K = 2 TO NCOL
2820 XSCM = XSCM + (Z(I, K) - YMIN)
2830 IF XSCM > XSCMAX THEN XSCMAX = XSCM
2840 NEXT K
2850 NEXT I
2860 XTC = XSCMAX / 10
2870 TCC = XTC: GOSUB 7100
2880 XTC = TCC: XM = 500
2890 GOTO 3000
2899 REM *** Setup 100% Stacked Bar Chart ***
2900 FOR I = 1 TO NROW: XSCM = 0
2910 FOR K = 2 TO NCOL
2920 XSCM = XSCM + (Z(I, K) - YMIN)
2930 ZP(I, K) = XSCM
2940 NEXT K
2950 FOR J = 2 TO NCOL
2960 ZD(I, J) = ZP(I, J) / XSCM
2970 NEXT J
2980 NEXT I
2990 XTC = 10
```

```
3000 GRAPH = 2: GOSUB 7800: REM Sets up Screen
3010 GOSUB 7500: DATBX = 1: REM Titles
3015 IF NSEL = 1 THEN J = VAL(DSET$): GOTO 3030
3020 FOR J = 1 TO NCOL - 1
3030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
3040 GOSUB 40000
3050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
3055 IF NSEL = 1 THEN 3070
3060 NEXT J
3070 IF VGRID$ = "N" THEN 3100
3080 GOSUB 7600: REM Draw Grid
3100 IF SCR% = 2 THEN CL(1) = 4 ELSE CL(1) = 1
3110 IF SCR% = 2 THEN CL(2) = 1 ELSE CL(2) = 1
3120 IF SCR% = 2 THEN CL(3) = 2 ELSE CL(3) = 1
3130 IF SCR% = 2 THEN CL(4) = 3 ELSE CL(4) = 1
3140 IF SCR% = 2 THEN CL(5) = 5 ELSE CL(5) = 1
3150 ON NSEL GOTO 3200, 3200, 3500, 3500
3200 FOR J = 1 TO NROW: IF NSEL = 1 THEN 3215
3210 FOR K = 2 TO NCOL
3215 IF NSEL = 1 THEN Y1 = (J - 1) * YTC + Y0 + 4
3220 IF NSEL = 2 THEN Y1 = (J - 1) * YTC + (K - 1) * YND + Y0 + 3
3225 IF DSET$ <> "*" THEN Y1 = (J - 1) * YTC + Y0 + 4
3230 X1 = X0 + 1
3240 Y2 = Y1 + (YND - 1)
3250 IF DSET$ <> "*" THEN Y2 = Y1 + YTC / 2
3255 X2 = (Z(J, K) - YMIN) * 50 / XTC + X0
3320 IF X1 > 560 THEN 3390
3330 IF Y1 > 170 THEN 3390
3340 IF SCR% <> 2 THEN 3360
3350 LINE (X1, Y1) - (X2, Y2), CL(K - 1), BF: GOTO 3380
3360 LINE (X1, Y1) - (X2, Y2), , B
3370 PAINT (X1 + 1, Y1 + 1), TILE$(K - 1)
3380 IF NSEL = 1 THEN 3400
3390 NEXT K
3400 NEXT J
3410 GOTO 9500
3499 REM ***** Stacked Bar Charts *****
3500 FOR J = 1 TO NROW
3510 X1 = X0 + 1
3520 FOR K = 2 TO NCOL
3530 Y1 = (J - 1) * YTC + Y0 + 3
3540 X1 = X1
3550 Y2 = J * YTC + Y0
3560 IF NSEL = 3 THEN X2 = X1 + (Z(J, K) - YMIN) * 50 / XTC
3570 IF NSEL = 4 THEN X2 = X0 + ZD(J, K) * 500
3580 IF X1 > 560 THEN 3650
3590 IF Y1 > 170 THEN 3650
3600 IF SCR% <> 2 THEN 3620
3610 LINE (X1, Y1) - (X2, Y2), CL(K - 1), BF: GOTO 3640
```

```
3620 LINE (X1, Y1) - (X2, Y2), , B
3630 PAINT (X1 + 2, Y1 + 2), TILE$(K - 1)
3640 X1 = X2
3650 NEXT K
3660 NEXT J
3670 GOTO 9500

7500 LNT = LEN(TITLE$): LNC = 40 - LNT/2
7510 LOCATE 2, LNC: PRINT TITLE$
7515 IF CH$ = "B" THEN 7525
7520 LNT = LEN(LXAXI$): LNC = 40 - LNT/2: GOTO 7530
7525 IF NSEL = 4 THEN LYAXI$ = "Percentage"
7526 LNT = LEN(LYAXI$): LNC = 40 - LNT/2
7527 LOCATE 21, LNC: PRINT LYAXI$: GOTO 7535
7530 LOCATE 21, LNC: PRINT LXAXI$
7535 IF YLAB$ = "V" THEN 7550
7536 IF NSEL = 4 THEN 7550
7537 IF CH$ = "B" THEN 7550
7540 LOCATE 4, 11: PRINT LYAXI$
7545 RETURN
7550 IF CH$ = "C" THEN YDIST$ = LYAXI$
7552 IF CH$ = "B" THEN YDIST$ = LXAXI$: GOTO 7560
7555 IF NSEL = 4 THEN YDIST$ = "Percentage"
7560 FOR I = 1 TO 15
7570 LOCATE 6 + I, 3: PRINT MID$ (YDIST$, I, 1)
7580 NEXT I
7590 RETURN
7800 IF SCR% <> 2 THEN 7820
7810 COLOR 15, 0
7820 CLS: GOSUB 9000
7830 IF SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
7840 LINE (10, 20) - (620, 180), RGB%, B
7850 IF GRAPH = 0 THEN RETURN
7860 IF SCR% = 2 THEN RGB% = 15 ELSE RGB% = 3
7870 LINE (X0, Y0 + YM) - (X0, Y0), RGB%: LINE - (X0 + XM, Y0), RGB%
7880 IF GRAPH = 1 THEN RETURN
7890 FOR LYD = 1 TO NROW
7900 AVGY = Y0 + 1 + INT(YTC / 2) + (LYD - 1) * YTC
7805 AVGX = X0 - 11: LDAT = Z(LYD, 1)
7910 PSET (X0, AVGY): LINE - (AVGX, AVGY), RGB%
7920 AVGX = AVGX - 20: AVGY = AVGY - 4: GOSUB 55000
7930 NEXT LYD
7940 FOR LXD = 0 TO 10
7950 AVGX = X0 + LXD * 50: AVGY - Y0 - 3: LDAT = YMIN + LXD * XTC
7960 PSET (AVGX, Y0): LINE - (AVGX, Y0 - 3), RGB%
7970 AVGX = AVGX - 9: AVGY = AVGY - 8: GOSUB 55000
7980 NEXT LXD
7990 RETURN
```

COMPUTER PROGRAM "GPLOT-C1"

To convert the program **"GRAPH-C1"** from screen graphics to an HPGL plotter the following changes are required:

replace segment (h) with the code for Lines 1000–1670, which are given at the end of this section
replace segment (k) with the code for Lines 3000–3670, which are given at the end of this section
replace segment (n) with Appendix A.28
replace segment (o) with the code for Lines 7500–7590, which are given at the end of this section
replace segment (p) with Appendix A.30
replace segment (q) with the code for Lines 7800–7990, which are given at the end of this section
replace segment (aa) with Appendix A.32
delete segments (v) and (bb)
add the following:

(cc) Appendix A.19	Line numbers 50000–50180
(dd) Appendix A.20	Line numbers 50500–50570
(ee) Appendix A.21	Line numbers 51000–51060
(ff) Appendix A.22	Line numbers 52000–52050
(gg) Appendix A.24	Line numbers 60000–60150

"GPLOT-C1"—BASIC program—segments (h), (k), (o) and (q) only

```
1000 GRAPH = 2: GOSUB 7300: REM Sets up Plotter
1010 GOSUB 7500: DATBX = 1: REM Titles
1015 GOSUB 50500: REM Header
1020 FOR J = 1 TO NCOL - 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 25
1031 X1 = AVGX - 4: Y1 = AVGY
1032 X2 = AVGX + 4: Y2 = AVGY + 4
1040 K = J + 1: GOSUB 52000
1045 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY; ";": GOSUB 60000
1050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1060 NEXT J
1150 ON NSEL GOTO 1200, 1200, 1500, 1500
1200 FOR J = 1 TO NROW: IF NSEL = 1 THEN 1215
1210 FOR K = 2 TO NCOL
1215 IF NSEL = 1 THEN X1 = (J - 1) * XTC + X0 + 15
1220 IF NSEL = 2 THEN X1 = (J - 1) * XTC + (K - 1) * XND + X0
1225 IF HIST$ = "Y" THEN X1 = (J - 1) * XTC + X0 + 1
1230 Y1 = Y0
1240 IF HIST$ = "N" THEN X2 = X1 + (XND - 1)
1245 IF DSET$ <> "*" THEN X2 = X1 + XTC / 2
1250 IF HIST$ = "Y" THEN X2 = X1 + XTC
1255 Y2 = (Z(J, K) - YMIN) * 12 / YTC + Y0
```

```
1320 IF X1 > 560 THEN 1390
1330 IF Y1 > 160 THEN 1390
1340 GOSUB 52000
1380 IF NSEL = 1 THEN 1400
1390 NEXT K
1400 NEXT J
1405 PRINT #3, "PU;SP0;": CLOSE
1410 GOTO 9530
1499 REM ***** Stacked Column Charts *****
1500 FOR J = 1 TO NROW
1510 Y1 = Y0
1520 FOR K = 2 TO NCOL
1530 X1 = (J - 1) * XTC + X0 + 10
1540 Y1 = Y1
1550 X2 = J * XTC + X0
1560 IF NSEL = 3 THEN Y2 = Y1 + (Z(J, K) - YMIN) * 12 / YTC
1570 IF NSEL = 4 THEN Y2 = Y0 + ZD(J, K) * 120
1580 IF X1 > 560 THEN 1650
1590 IF Y1 > 160 THEN 1650
1600 GOSUB 52000
1640 Y1 = Y2
1650 NEXT K
1660 NEXT J
1605 PRINT #3, "PU;SP0;": CLOSE
1670 GOTO 9530

3000 GRAPH = 2: GOSUB 7800: REM Sets up Screen
3010 GOSUB 7500: DATBX = 1: REM Titles
3015 GOSUB 50500: REM Header
3020 FOR J = 1 TO NCOL - 1
3030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
3031 X1 = AVGX - 4: Y1 = AVGY
3032 X2 = AVGX + 4: Y2 = AVGY + 4
3040 K = J + 1: GOSUB 52000
3045 PRINT #3, "PU;PA"; AVGX + 10; "," AVGY; ";": GOSUB 60000
3050 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
3060 NEXT J
3150 ON NSEL GOTO 3200, 3200, 3500, 3500
3200 FOR J = 1 TO NROW: IF NSEL = 1 THEN 3215
3210 FOR K = 2 TO NCOL
3215 IF NSEL = 1 THEN Y1 = (J - 1) * YTC + Y0 + 4
3220 IF NSEL = 2 THEN Y1 = (J - 1) * YTC + (K - 1) * YND + Y0 + 3
3225 IF DSET$ <> "*" THEN Y1 = (J - 1) * YTC + Y0 + 4
3230 X1 = X0
3240 Y2 = Y1 + (YND - 1)
3250 IF DSET$ <> "*" THEN Y2 = Y1 + YTC / 2
3255 X2 = (Z(J, K) - YMIN) * 50 / XTC + X0
3320 IF X1 > 560 THEN 3390
3330 IF Y1 > 170 THEN 3390
```

```
3340 GOSUB 52000
3380 IF NSEL = 1 THEN 3400
3390 NEXT K
3400 NEXT J
3405 PRINT #3, "PU;SP0;": CLOSE
3410 GOTO 9530
3499 REM ***** Stacked Bar Charts *****
3500 FOR J = 1 TO NROW
3510 X1 = X0
3520 FOR K = 2 TO NCOL
3530 Y1 = (J - 1) * YTC + Y0 + 3
3540 X1 = X1
3550 Y2 = J * YTC + Y0
3560 IF NSEL = 3 THEN X2 = X1 + (Z(J, K) - YMIN) * 50 / XTC
3570 IF NSEL = 4 THEN X2 = X0 + ZD(J, K) * 500
3580 IF X1 > 560 THEN 3650
3590 IF Y1 > 160 THEN 3650
3600 GOSUB 52000
3640 X1 = X2
3650 NEXT K
3660 NEXT J
3605 PRINT #3, "PU;SP0;": CLOSE
3670 GOTO 9530

7500 LNT = LEN(TITLE$)
7501 PRINT #3, "SP2;SI0.3,0.55;SL0.2;": GOSUB 60000
7502 PRINT #3, "PU;PA320,185;": GOSUB 60000
7503 PRINT #3, "CP"; -LNT / 2; "0;": GOSUB 60000
7510 PRINT #3, "LB"; TITLE$; CHR$(3): GOSUB 60000
7511 PRINT #3, "SP1;SL;SI0.2,0.3;PU;PA320,35;": GOSUB 60000
7515 IF CH$ = "B" OR CH$ = "b" THEN 7525
7520 LNT = LEN(LXAXI$): PRINT #3, "CP"; -LNT / 2; "0;": GOSUB 60000
7521 GOTO 7530
7525 LNT = LEN(LYAXI$): PRINT #3, "CP"; -LNT/2; "0;": GOSUB 60000
7526 PRINT #3, "LB"; LYAXI$; CHR$(3): GOSUB 60000: GOTO 7550
7530 PRINT #3, "LB"; LXAXI$; CHR$(3): GOSUB 60000
7550 IF CH$ = "C" OR CH$ = "c" THEN YDIST$ = LYAXI$
7555 IF CH$ = "B" OR CH$ = "b" THEN YDIST$ = LXAXI$
7560 LNC = LEN(YDIST$)
7570 PRINT #3, "PU;PA30,65;": GOSUB 60000
7575 PRINT #3, "DI 0,1;": GOSUB 60000
7580 PRINT #3, "LB"; YDIST$; CHR$(3): GOSUB 60000
7585 PRINT #3, "DI 1,0;"
7590 RETURN
7800 GOSUB 50000
7805 PRINT #3, "IN;"
7810 PRINT #3, "IP"; IPXMIN; ","; IPYMIN; ","; IPXMAX; ","; IPYMAX;
   ";"
7815 PRINT #3, "SC 0,640,0,200;"
```

```
7820 PRINT #3, "SP2;PU;PA 10,20;PD;PA 620,20;": GOSUB 60000
7825 PRINT #3, "PA 620,180;": GOSUB 60000
7830 PRINT #3, "PA 10,180;": GOSUB 60000
7835 PRINT #3, "PA 10,20;PU;": GOSUB 60000
7850 IF GRAPH = 0 THEN RETURN
7860 PRINT #3, "SP1;PA"; X0; ","; Y0 + YM; ";": GOSUB 60000
7865 PRINT #3, "PD;PA "; X0; ","; Y0; ";": GOSUB 60000
7870 PRINT #3, "PA "; X0 + XM; ","; Y0; ";": GOSUB 60000
7880 IF GRAPH = 1 THEN RETURN
7885 PRINT #3, "SI0.2,0.3;"
7890 FOR LYD = 1 TO NROW
7900 AVGY = Y0 + INT(YTC / 2) + (LYD - 1) * YTC
7905 AVGX = X0 - 11: LDAT = Z(LYD, 1)
7910 PRINT #3, "PU "; X0; ","; AVGY; ";YT;": GOSUB 60000
7920 PRINT #3, "PU "; X0; ","; AVGY; ";CP-4.5,-0.25;"
7925 PRINT #3, "LB "; LDAT; CHR$(3)
7930 NEXT LYD
7940 FOR LXD = 0 TO 10
7950 AVGX = X0 + LXD * 50: AVGY = Y0 - 3: LDAT = YMIN + LXD * XTC
7960 PRINT #3, "PU"; AVGX; ","; Y0; ";XT;": GOSUB 60000
7970 PRINT #3, "PU"; AVGX; ","; Y0; ";CP-2,-l; LB"; LDAT; CHR$(3)
7980 NEXT LXD
7990 RETURN
```

Appendix D
Pie charts and contour maps

APPENDIX D.1:

PIE CHARTS COMPUTER PROGRAM "GRAPH-D1"

This program was discussed in Section 4.2, and covers both single and dual pie charts.

Pie charts

(a) Initialization and control line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables which require to be initialized for use in the program. Note the radius **(R1)** of the pie chart is set to **80**.

(b) Data loading Line numbers 100–195

See the notes for section (b) of Appendix B.1. If the file chosen has too many data points for a pie chart display, this will be noted on screen and the program returned to the menu to allow the user to change the data file.

(c) Pie chart set-up Line numbers 200–440

This segment of code calculates the percentages of the pie slices and then selects the colours **CL%(I)** for each pie slice. An opportunity is also given to change the graph titles.

(d) Text file entry Line numbers 500–590

See the notes for section (d) of Appendix B.1.

(e) Data set check Line numbers 600–690

This routine selects which data set to display in a multiple data set, or gives the user an opportunity to display single or dual pie charts if there are only two data sets.

(f) Pie chart presentation Line numbers 700–940

The user now has to choose whether to present the data values or percentages alongside each pie slice. A pie chart caption can now be entered, and this is essential if displaying dual pie charts to identify the pies. Finally the user has a choice of exploding the first pie segment.

(g) Pie chart display — Line numbers 1000–1540

This is the main part of the graphics display and sets up the screen with graph titles and data identifiers. Note that a ratio **ASPECT** is defined in Line 1100. This is to give circles on the screen rather than ellipses. The user may have to change this ratio if their pie display is not circular. After the display of a single or dual pie chart, the user can obtain a dump by pressing the [Shift] and [Prt sc] keys simultaneously.
This program also requires the following routines.

(h) Appendix A.10	Line numbers 7300–7490
(i) Appendix A.11	Line numbers 7500–7590
(j) Appendix A.2	Line numbers 8000–8180
(k) Appendix A.3	Line numbers 8200–8390
(l) Appendix A.4	Line numbers 8400–8490
(m) Appendix A.5	Line numbers 8500–8590
(n) Appendix A.14	Line numbers 8600–8660
(o) Appendix A.3	Line numbers 8800–8990
(p) Appendix A.15	Line numbers 9000–9050
(q) Appendix A.16	Line numbers 9500–9600
(r) Appendix A.17	Line numbers 10000–10310
(s) Appendix A.18	Line numbers 40000–40590
(t) Appendix A.19	Line numbers 55000–56960

“GRAPH-D1”—BASIC program

```
10 REM <GRAPH-D1> Presentation Graphics Program D.1 - Pie Charts
14 REM (C) Copyright P.H.Milne 1990
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: DUAL = 0: DSET = 1: PIE = 1
70 Y0 = 50: A$ = "*": DATBX = 1: R1 = 80: A2 = 0: PI = 3.1415926#
80 DIM CL%(10), Z(10, 6), ZP(10, 6), DAT$(10)
90 WINDOW (0, 0) - (639, 199)
100 CLS: PC$ = "Pie Chart"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
150 A$ = INPUT$(1)
160 IF INSTR("YNyn", A$) = 0 THEN 140
```

```
170 IF A$ = "N" OR A$ = "n" THEN 9500
180 IF SCR% = 2 AND NROW < 11 THEN 200
185 IF NROW > 5 THEN LOCATE 24, 10: PRINT "Too Many DATA Points";
190 IF NROW > 5 THEN 9600
195 GOSUB 8600
199 REM **** Set Up Percentage Pie ****
200 FOR K = 2 TO NCOL: DTOT = 0
210 FOR I = 1 TO NROW
220 DTOT = DTOT + Z(I, K)
230 NEXT I
240 FOR J = 1 TO NROW
250 ZP(J, K) = Z(J, K) / DTOT
260 NEXT J
270 NEXT K
300 IF SCR% = 2 THEN CL%(1) = 4 ELSE CL%(1) = 1
310 IF SCR% = 2 THEN CL%(2) = 1 ELSE CL%(2) = 1
320 IF SCR% = 2 THEN CL%(3) = 5 ELSE CL%(3) = 1
330 IF SCR% = 2 THEN CL%(4) = 3 ELSE CL%(4) = 1
340 IF SCR% = 2 THEN CL%(5) = 2 ELSE CL%(5) = 1
350 IF SCR% = 2 THEN CL%(6) = 13 ELSE CL%(6) = 1
360 IF SCR% = 2 THEN CL%(7) = 9 ELSE CL%(7) = 1
370 IF SCR% = 2 THEN CL%(8) = 6 ELSE CL%(8) = 1
380 IF SCR% = 2 THEN CL%(9) = 10 ELSE CL%(9) = 1
390 IF SCR% = 2 THEN CL%(10) = 11 ELSE CL%(10) = 1
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 21, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS : GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 YLAB$ = "V"
545 IF PIE = 1 THEN NR = NROW ELSE NR = NCOL • 1
550 FOR I = 1 TO NR
560 LOCATE 11 + I, 10:
561 PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 IF ND = 1 THEN DSET = 1: GOTO 700
610 IF ND > 2 THEN 660
620 LOCATE 22, 10: PRINT "Display Both Data Sets (Y/N)";
```

```
625 DUP$ = INPUT$(1)
630 IF INSTR("YNyn", DUP$) = 0 THEN 620
640 IF DUP$ = "y" THEN DUP$ = "Y"
645 IF DUP$ = "n" THEN DUP$ = "N"
650 IF DUP$ = "Y" THEN 700
660 LOCATE 23, 10: PRINT "Which Data Set to plot 1- "; ND;
665 DSET$ = INPUT$(1)
670 IF INSTR( "12345", DSET$) = 0 THEN 660
680 DSET = VAL(DSET$)
690 IF ND < DSET THEN 660
700 CLS : GOSUB 9000
710 LOCATE 5, 10: PRINT "Select Pie Chart Presentation :-"
720 LOCATE 8, 10: PRINT "<1> Display Pie by Value"
730 LOCATE 10, 10: PRINT "<2> Display Pie by Percentage"
740 LOCATE 12, 10: PRINT "<3> Return to Menu"
750 LOCATE 16, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
760 IF INSTR("123", SEL$) = 0 THEN 750
770 NSEL = VAL(SEL$)
780 IF NSEL = 1 THEN LYAXI$ = "Value"
790 IF NSEL = 2 THEN LYAXI$ = "Percentage"
800 LOCATE 20, 10: PRINT "Enter Pie Caption No. 1";
810 INPUT CAPT$(DSET)
820 IF DUP$ <> "Y" THEN 900
830 LOCATE 22, 10: PRINT "Enter Pie Caption No. 2";
840 INPUT CAPT$(2)
900 LOCATE 24, 10: PRINT "Explode first Segment (Y/N) ";
910 PEX$ = INPUT$(1)
920 IF INSTR("YNyn", PEX$) = 0 THEN 900
930 IF PEX$ = "y" THEN PEX$ = "Y"
940 IF PEX$ = "n" THEN PEX$ = "N"
1000 GRAPH = 0: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: DATBX = 1: IF NROW > 5 THEN NR = 5 : GOTO 1030
1020 IF PIE = 1 THEN NR = NROW ELSE NR = NCOL • 1
1030 FOR J = 1 TO NR
1040 AVGX = 94 + 103 * (J • 1): AVGY = Y0 • 22
1050 GOSUB 40000
1060 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1070 NEXT J
1080 IF DUP$ = "Y"
1090 CY = 100
1100 IF SCR% = 2 THEN ASPECT
1199 REM **** Single Pie ****
1200 FOR C = 1 TO NROW
1210 A1 = A2: IF C = 1 THEN A1 = .0001
1220 A2 = A2 + ZP(C, DSET + 1) * 2 * PI
```

```
1230 AA = (A1 + A2) / 2
1240 IF C = 1 AND PEX$ = "Y" THEN PCX = CX + 15 ELSE PCX = CX
1250 IF C = 1 AND PEX$ = "Y" THEN PCY = CY + 5 ELSE PCY = CY
1260 CIRCLE (PCX, PCY), R1 , 15, • A1 , • A2 , ASPECT
1270 PSET (PCX + COS(AA) * 1 * R1 , PCY + SIN(AA) * .4 * R1)
1280 LINE •  (PCX + COS(AA) * 1.5 * R1 , PCY + SIN(AA) * .6 * R1) , 15
1290 AVGX = PCX + COS(AA) * 1.5 * R1 : AVGY = PCY + SIN(AA) * . 6 * R1
1300 IF AVGY < 100 THEN AVGY = AVGY •  10
1310 IF NSEL = 1 THEN LDAT = Z(C, DSET + 1)
1320 IF NSEL = 2 THEN LDAT = INT(ZP(C, DSET + 1) * 100 + .5)
1330 GOSUB 55000
1390 IF SCR% <> 2 THEN 1420
1400 CTX = COS(AA) * .5 * R1 : STY = SIN(AA) * . 2 * R1
1405 PAINT (PCX + CTX, PCY + STY), CL%(C), 15
1410 GOTO 1430
1420 PAINT (PCX + CTX, PCY + STY), TILE$(C)
1430 NEXT C
1440 LNT = LEN(CAPT$(DSET)): LNC = CX / 8 + 1 •  LNT / 2
1450 LOCATE 5, LNC: PRINT CAPT$(DSET)
1460 IF DUP$ = "Y" THEN 1500
1470 GOTO 9500
1499 REM ***** DUAL PIE *****
1500 IF DUAL = 1 THEN 9500
1510 DSET = DSET + 1
1520 CX = 440
1530 A2 = 0: C = 0: DUAL = 1
1540 GOTO 1200
```

COMPUTER PROGRAM "GPLOT-D1"

To convert the program **"GRAPH-D1"** from screen graphics to an HPGL plotter the following changes are required:

replace segment (g) with the code for Lines 1000–1540, which are given at the end of this section
- replace segment (h) with Appendix A.28—note change of scaling factors in Line 7315
- replace segment (i) with Appendix A.29
- replace segment (s) with Appendix A.32
- delete segments (n) and (t)
- add the following:

(u) Appendix A.19	Line numbers 50000–50180
(v) Appendix A.20	Line numbers 50500–50570
(w) Appendix A.22	Line numbers 52000–52040
(x) Appendix A.24	Line numbers 60000–60150

"GPLOT-D1"—BASIC program—segment (g) only

```
1000 GRAPH = 0: GOSUB 7300: REM Sets up Screen
1005 GOSUB 50500: REM Header
1010 GOSUB 7500: DATBX = 1: IF NROW > 5 THEN NR: GOTO 1030
1020 IF PIE = 1 THEN NR = NROW ELSE NR = NCOL •  1
1030 FOR J = 1 TO NR
1040 AVGX = 94 + 103 * (J •  1): AVGY = Y0 •  22 * MX
1050 GOSUB 40000
1055 PRINT #3, "PU;PA"; AVGX + 10; ","; AVGY •  2 * MX; ";"
1056 GOSUB 60000
1060 PRINT #3, "LB"; DAT$(J); CHR$(3): GOSUB 60000
1070 NEXT J
1080 IF DUP$ = "Y" THEN CX = 160 ELSE CX = 300
1090 CY = 100 * MX
1100 IF SCR% = 2 THEN ASPECT = 5 / 6 ELSE ASPECT = 5 / 12
1199 REM **** Single Pie ****
1200 FOR C = 1 TO NROW
1205 SB = 360 * ZP(C, DSET + 1)
1210 A1 = A2: IF C = 1 THEN A1 = .0001
1220 A2 = A2 + ZP(C, DSET + 1) * 2 * PI
1230 AA = (A1 + A2) / 2
1240 IF C = 1 AND PEX$ = "Y" THEN PCX = CX + 10 ELSE PCX = CX
1250 IF C = 1 AND PEX$ = "Y" THEN PCY = CY + 5 * MX ELSE PCY = CY
1255 GOSUB 52000
1260 PRINT #3, "PU;PA"; PCX; ","; PCY; ";": GOSUB 60000
1262 PRINT #3, "EW"; R1; ", "; SA; ","; SB; ";": GOSUB 60000
1265 PRINT #3, "WG"; R1; ","; SA; ","; SB; ";": GOSUB 60000
1266 CTX = COS(AA) * 1 * R1: STY = (SIN(AA) * .4 * R1)
1270 PRINT #3, "PU;PA"; PCX + CTX; ","; PCY + STY * MX; ";"
1275 CTX = CTX * 1.5: STY = STY * 1.5
1280 PRINT #3, "PD;PA"; PCX + CTX; ","; PCY + STY * MX; ";"
1290 AVGX = PCX + COS(AA) * 1.5 * R1
1291 AVGY = PCY + (SIN(AA) * .6 * R1) * MX
1295 IF AVGX < PCX THEN AVGX = AVGX •  30
1300 IF AVGY < 200 THEN AVGY = AVGY •  10
1310 IF NSEL = 1 THEN LDAT = Z(C, DSET + 1)
1320 IF NSEL = 2 THEN LDAT = INT(ZP(C, DSET + 1) * 100 + .5)
1330 PRINT #3, "PU;PA"; AVGX; ","; AVGY; ";": GOSUB 60000
1340 PRINT #3, "LB"; LDAT; CHR$(3): GOSUB 60000
1350 SA = SA + SB
1430 NEXT C
1440 PRINT #3, "PU;PA"; PCX; ",320;": GOSUB 60000
1450 LNT = LEN(CAPT$(DSET))
1451 PRINT #3, "CP"; •  LNT / 2; "0;": GOSUB 60000
1455 PRINT #3, "SP2;LB"; CAPT$(DSET); CHR$(3): GOSUB 60000
1460 IF DUP$ = "Y" THEN 1500
```

```
1465 PRINT #3, "PU;PA0,0;SP0;": CLOSE #3
1470 GOTO 9530
1499 REM ***** DUAL PIE *****
1500 IF DUAL = 1 THEN 1465
1510 DSET = DSET + 1
1520 CX = 440
1530 A2 = 0: C = 0: DUAL = 1
1540 GOTO 1200
```

APPENDIX D.2:
LINE CONTOURS COMPUTER PROGRAM "GRAPH-D2"

This program was discussed in Section 4.3.1 and is suitable for drawing contours either at user selected intervals in highlighted colours, or at regular contour intervals in a 'rainbow' spectrum from the lowest (blue) to the highest (magenta). Readers of my previous book, *Computer Graphics for Surveying* will note this program is an enhanced version of PC-Contour.

Line contours

(a) Initialization and control — Line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables which require to be initialized for use in the program. For memory considerations, note that the maximum number of grid points, i.e. rows and columns is 50 each, **Z(50,50)**. Another array **C(15)** is used for the rainbow colours.

(b) Data loading — Line numbers 100–160

See the notes for section (b) of Appendix B.1. If the file found in the routine at Line 8200 is not a contour **('.CON')** or generated surface file **('.EQD'),** the user is advised accordingly and given the opportunity to load a suitable file. The number of rows and columns of the chosen file are displayed on screen.

(c) Line contour set-up — Line numbers 200–460

This segment of code displays the maximum and minimum *X, Y* and *Z* values in the data file and gives the user the option to plot either (A)ll the contours (i.e. in 'rainbow' colours) or (S)elected contours at specified levels. An option is also available to plot a background grid for reference.

(d) Line contour display — Line numbers 3000–4050

This is the main part of the program. The screen is set up as for other presentation graphics programs using the three **GOSUB** routines at Lines 7300, 7500 and 7600. If (A)ll the contours are to be plotted, then a colour shaded panel is drawn at the right-hand side of the screen to identify the contour levels (this was described in Appendix A.13). If (S) elected contours are to be drawn, the user has to enter the highest **(HC)** and lowest **(LC)** contours and the contour interval **(CI)**. At the end of the plot, if (S)elected contours was chosen, the user has an opportunity to add (M)ore contours or to (E)rase the screen and start again with other levels.

This program also requires the following routines—note those shown with an * require some changes to the standard routines listed in Appendix A to cater for contours.

(e) Appendix A.9	Line numbers 7100–7290
(f) Appendix A.10	Line numbers 7300–7490
(g) Appendix A.11	Line numbers 7500–7590
(h) Appendix A.12	Line numbers 7600–7710
(i) Appendix A.13	Line numbers 7800–7910
(j) Appendix A.2	Line numbers 8000–8180
(k) Appendix A.3*	Line numbers 8200–8390
(l) Appendix A.5	Line numbers 8500–8590
(m) Appendix A.3*	Line numbers 8800–8990
(n) Appendix A.15	Line numbers 9000–9050
(o) Appendix A.16	Line numbers 9500–9600
(p) Appendix A.17	Line numbers 10000–10310
(q) Appendix A.19	Line numbers 55000–56960

"GRAPH-D2"—BASIC program

```
10 REM <GRAPH-D2> Presentation Graphics Program D.2 - Line Contours
14 REM (C) Copyright P.H.Milne 1990
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: **** Maximum 50,50 ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
70 GX0 = 70: GY0 = 50: DATBX = 0: SR% = 2: SM = 2.4
75 A$ = "*": CSEL$ = "S"
80 DIM Z(50, 50), C(15)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Line Contours"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
200 CLS : GOSUB 9000
210 LOCATE 6, 10: PRINT "Contour DATA File "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
```

```
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
280 IF SCR% <> 2 THEN 340
290 LOCATE 16, 10: PRINT "Choose (A)ll or (S)elected Contours";
295 CSEL$ = INPUT$(1)
300 IF INSTR("ASas", CSEL$) = 0 THEN 290
310 IF CSEL$ = "a" THEN CSEL$ = "A"
320 IF CSEL$ = "s" THEN CSEL$ = "S"
340 XTC = (XMAX •  XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 500
370 YTC = (YMAX •  YMIN) / 10
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 120
400 LOCATE 20, 10: PRINT "Background Grid (Y/N) "; : A$ = INPUT$(1)
410 IF INSTR("YNyn", A$) = 0 THEN 400
420 IF A$ = "y" THEN A$ = "Y"
430 IF A$ = "n" THEN A$ = "N"
440 IF A$ = "Y" THEN HGRID = 1: VGRID = 1: YLAB$ = "V"
450 IF A$ = "N" THEN HGRID = 0: VGRID = 0: YLAB$ = "V"
460 GOSUB 8500
2995 REM ***** Screen Plot *****
3000 GRAPH = 2: SX = 240 * SR%: SY = 120
3010 GX = GX0: GY = GY0: MG = 0: MH = 0: MV = 0: PP = 0
3020 MH = SX / ((NCOL •  1) * SM): MV = SY / (NROW •  1)
3030 IF MH <= MV THEN MG = MH ELSE MG = MV
3050 IF SCR% = 1 THEN RGB% = 2: CL% = 1
3051 IF SCR% = 1 AND SD% = 2 THEN RGB% = 3: CL% = 1
3052 IF SCR% = 2 THEN RGB% = 15: CL% = 14
3053 IF SCR% = 3 THEN RGB% = 3: CL% = 1
3054 IF SCR% = 4 THEN RGB% = 3: CL% = 1
3060 PX = GX + (MG * (NCOL •  1) * SM) : PY = GY + MG * (NROW •  1)
3070 GOSUB 7300: GOSUB 7500: REM Set up Screen
3075 IF HGRID = 0 THEN 3090
3080 GOSUB 7600: REM Draw Grid
3090 LINE (GX, GY) •  (PX, PY), RGB%, B
3095 IF PP = 1 THEN 3880
3099 REM *** Set up Colour Shading ***
3100 IF CSEL$ = "S" THEN 3200
3110 NOCL = 13
3120 C$ = "1314051204101102030901060815"
3130 FOR K = 0 TO NOCL
3140 C(K) = VAL(MID$(C$, K * 2 + 1, 2))
3150 NEXT K
3160 TCC = (ZMAX •  ZMIN) / (NOCL + 1)
3170 GOSUB 7100: CI = TCC
```

```
3180 HC = INT(ZMAX / TCC) * TCC: LC = INT(ZMIN / TCC) * TCC
3190 NOCL = (HC • LC) / TCC: GOSUB 7800
3200 IF CSEL$ = "A" THEN 3300
3240 LOCATE 4, 32 * SR%: PRINT "CONTOURS"
3250 LOCATE 5, 32 * SR%: PRINT "LOW VAL:"
3252 LOCATE 6, 32 * SR%: PRINT " "
3255 LOCATE 6, 32 * SR%: INPUT LC
3256 IF LC < ZMIN THEN 3252
3257 IF LC > ZMAX THEN 3252
3260 LOCATE 7, 32 * SR%: PRINT "HIGH VAL:"
3262 LOCATE 8, 32 * SR%: PRINT " "
3265 LOCATE 8, 32 * SR%: INPUT HC
3266 IF HC > ZMAX THEN 3262
3267 IF HC < ZMIN THEN 3262
3268 IF HC < LC THEN 3262
3270 LOCATE 9, 32 * SR%: PRINT "CONTOUR"
3272 LOCATE 10, 32 * SR%: PRINT " "
3275 LOCATE 10, 32 * SR%: INPUT "INT"; CI
3276 IF CI <= 0 THEN 3272
3300 FOR CC = LC TO HC STEP CI
3301 IF CSEL$ = "A" THEN 3320
3302 LOCATE 11, 32 * SR%: PRINT "CURRENT"
3304 LOCATE 12, 32 * SR%: PRINT " "
3306 LOCATE 12, 32 * SR%: PRINT "="; CC
3310 IF SCR% > 2 THEN 3319
3311 IF SCR% = 2 THEN 3315
3312 IF CC / 10 = INT(CC / 10) THEN CL% = 3: GOTO 3320
3313 IF (CC • 5) / 10 = INT((CC • 5) / 10) THEN CL% = 1: GOTO 3320
3314 CL% = 2: GOTO 3320
3315 IF CC / 10 = INT(CC/10) THEN CL% = 12: GOTO 3320
3316 IF (CC • 5) / 10 = INT((CC • 5) / 10) THEN CL% = 10: GOTO 3320
3317 CL% = 14: GOTO 3320
3319 CL% = 1
3320 FOR I = 0 TO (NROW • 2)
3330 Y0 = MG * I
3340 FOR J = 0 TO (NCOL • 2)
3350 X0 = MG * J
3360 NP = 0: CLOR = INT((HC • CC) / TCC)
3370 Z1 = Z(I, J)
3380 IF CC > Z1 THEN NP = NP + 1
3390 Z2 = Z(I, (J + 1))
3400 IF CC > Z2 THEN NP = NP + 1
3410 Z3 = Z((I + 1), J)
3420 IF CC > Z3 THEN NP = NP + 1
3430 Z4 = Z((I + 1), (J + 1))
```

```
3440 IF CC > Z4 THEN NP = NP + 1
3450 IF NP = 0 OR NP = 4 GOTO 3840
3460 A = Z1
3470 B = Z2 •  A
3480 C = Z3 •  A
3490 D = Z4 •  A •  B •  C
3500 ZT = 0
3510 FOR Y1 = 0 TO 1 STEP .25
3520 DR = B + D * Y1
3530 IF DR = 0 GOTO 3680
3540 X1 = (CC •  A •  C * Y1) / DR
3550 IF X1 < 0 OR X1 > 1 GOTO 3680
3560 X = X0 + MG * X1
3570 Y = Y0 + MG * Y1
3580 IF X > PX OR Y > PY GOTO 3670
3590 IF ZT = 0 THEN 3615
3595 IF CSEL$ = "A" THEN 3605
3600 LINE •  ((X * SM + GX), (Y + GY)), CL%: GOTO 3610
3605 LINE •  ((X * SM + GX), (Y + GY)), C(CLOR)
3610 GOTO 3670
3615 IF CSEL$ = "A" THEN 3625
3620 PSET ((X * SM + GX), (Y + GY)) , CL%: GOTO 3630
3625 PSET ((X * SM + GX), (Y + GY)), C(CLOR)
3630 IF TL > 0 THEN 3670
3631 IF CC / 5 <> INT(CC / 5) THEN 3670
3632 LDAT = CC
3633 AVGX = (X * SM + GX)
3634 AVGY = (Y + GY)
3640 GOSUB 55000
3650 TL = TL + 1
3660 GOTO 3615
3670 ZT = ZT + 1
3680 NEXT Y1
3690 ZT = 0
3700 FOR X1 = 0 TO 1 STEP .25
3710 DS = C + D * X1
3720 IF DS = 0 GOTO 3830
3730 Y1 = (CC •  A •  B * X1) / DS
3740 IF Y1 < 0 OR Y1 > 1 GOTO 3830
3750 X = X0 + MG * X1
3760 Y = Y0 + MG * Y1
3770 IF X > PX OR Y > PY GOTO 3820
3780 IF ZT = 0 THEN 3805
3785 IF CSEL$ = "A" THEN 3795
3790 LINE •  ((X * SM + GX), (Y + GY)), CL%: GOTO 3800
```

```
3795 LINE • ((X * SM + GX), (Y + GY)), C(CLOR)
3800 GOTO 3820
3805 IF CSEL$ = "A" THEN 3815
3810 PSET ((X * SM + GX), (Y + GY)), CL%: GOTO 3820
3815 PSET ((X * SM + GX), (Y + GY)), C(CLOR)
3820 ZT • ZT + 1
3830 NEXT X1
3840 NEXT J
3850 NEXT I
3860 TL = 0
3870 NEXT CC
3875 PP = 1: GOTO 3090
3880 Z$ = INPUT$(1)
3890 IF Z$ = " " THEN 3880 ELSE 3900
3900 IF CSEL$ = "A" THEN 9530
3910 LOCATE 16, 32 * SR% : PRINT "'(M)ORE'"
3920 LOCATE 17, 32 * SR% : PRINT "CONTOURS"
3930 LOCATE 18, 32 * SR% : PRINT "' (E)RASE' ,"
3940 LOCATE 19, 32 * SR% : PRINT "or '(Q)uit'"
3945 BEEP
3950 LOCATE 20, 32 * SR% : PRINT " "
3955 LOCATE 20, 32 * SR% : PRINT "?"; : M$ = INPUT$(1)
3960 IF INSTR("MEQmeq", M$) = 0 THEN 3945
3965 IF M$ = "m" THEN M$ = "M"
3970 IF M$ = "e" THEN M$ = "E"
3975 IF M$ = "q" THEN M$ = "Q"
3980 FOR Q = 1 TO 5
3990 LOCATE 15 + Q, 32 * SR% : PRINT " "
4000 NEXT Q
4010 IF M$ = "M" THEN 3200
4020 IF M$ = "E" THEN 3000
4030 IF M$ = "Q" THEN 4040
4040 M$ = " ": SR% = 1
4050 GOTO 9530

8200 CLS: GOSUB 9000
8210 LOCATE 6, 10: PRINT "Please wait ..... finding File - "
8220 OPEN "I", #2, DDSK$ + "IDFILE.PGD"
8230 INPUT #2, DFILEN$
8240 INPUT #2, EXT$
8250 CLOSE #2
8251 IF EXT$ = ".EQD" THEN 8260
8252 IF EXT$ = ".CON" THEN 8260
8253 LOCATE 8, 10: PRINT "This Program requires a Contour data File"
8255 GOTO 8800
8260 LOCATE 6, 45: PRINT DFILEN$
```

```
8270 LOCATE 8, 10: PRINT "Load DATA for above File (Y/N)";
8271 A$ = INPUT$(1)
8275 IF INSTR("YNyn", A$) = 0 THEN 8270
8280 IF A$ = "N" OR A$ = "n" THEN 8800
8290 LOCATE 10, 10: PRINT "Please wait ..... reading DATA"
8300 OPEN "I", #1, DDSK$ + DFILEN$ + EXT$
8310 INPUT #1, NROW, NCOL
8315 INPUT #1, ZMAX, ZMIN: IF EXT$ = ".CON" THEN 8325
8320 INPUT #1, XMIN, XMAX, YMIN, YMAX: GOTO 8330
8325 INPUT #1, DX, DY: XMIN = 0: XMAX = DX: YMIN = 0: YMAX = DY
8330 FOR I = (NROW - 1) TO 0 STEP -1
8340 FOR J = 0 TO (NCOL - 1)
8350 INPUT #1, Z(I, J)
8360 NEXT J
8370 NEXT I
8380 CLOSE #1
8390 RETURN

8800 LOCATE 10, 10: PRINT "Select Extension"
8810 LOCATE 10, 30: PRINT "<1> '.CON' Contour Data File"
8820 LOCATE 12, 30: PRINT "<2> '.EQD' Surface Data File"
8830 LOCATE 14, 30: PRINT "<3> Return to Menu "; : A$ = INPUT$(1)
8840 IF INSTR("123", A$) = 0 THEN 8830
8850 NSEL = VAL(A$)
8860 ON NSEL GOTO 8870, 8880, 9600
8870 EXT$ = ".CON": DESP$ = "Contour": GOTO 8900
8880 EXT$ = ".EQD": DESP$ = "Surface"
8900 CLS : DFILEN$ = " ": GOSUB 9000: PRINT : FILES DDSK$ + "*" +
   EXT$
8910 LOCATE 15, 5: PRINT "Type"; DESP$; "Model File Name"
8920 LOCATE 17, 5: INPUT "(Type 'QUIT' to Return to Menu)"; DFILEN$
8930 IF DFILEN$ = "QUIT" OR DFILEN$ = "quit" THEN 9600
8940 OPEN "O", #2, DDSK$ + "IDFILE.PGD"
8950 PRINT #2, DFILEN$
8960 PRINT #2, EXT$
8970 CLOSE #2
8980 CLS : GOSUB 9000
8990 GOTO 8290
```

COMPUTER PROGRAM “GPLOT-D2”

To convert the program **“GRAPH-D2”** from screen graphics to an HPGL plotter the following changes are required:

replace segment (d) with the code for Lines 3000–4050, which are given at the end of this section
- replace segment (f) with Appendix A.28
- replace segment (g) with Appendix A.29
- replace segment (h) with Appendix A.30
- delete segments (i) and (q)

add the following:

(r) Appendix A.19 Line numbers 50000–50180
(s) Appendix A.20 Line numbers 50500–50570
(t) Appendix A.24 Line numbers 60000–60150

GPLOT-D2—BASIC program—segment (d) only

```
3000 GRAPH = 2: SX = 240 * SR%: SY = 120
3010 GX = GX0: GY = GY0: MG = 0: MH = 0: MV = 0: PP = 0
3020 MH = SX / ((NCOL - 1) * SM): MV = SY / (NROW - 1)
3030 IF MH <= MV THEN MG = MH ELSE MG = MV
3050 IF SCR% = 1 THEN RGB% = 2: CL% = 1
3051 IF SCR% = 1 AND SD% = 2 THEN RGB% = 3: CL% = 1
3052 IF SCR% = 2 THEN RGB% = 15: CL% = 14
3053 IF SCR% = 3 THEN RGB% = 3: CL% = 1
3054 IF SCR% = 4 THEN RGB% = 3: CL% = 1
3060 PX = GX + (MG * (NCOL - 1) * SM): PY = GY + MG * (NROW - 1)
3070 GOSUB 7300: GOSUB 7500: GOSUB 50500: REM Set up Plotter
3075 IF HGRID = 0 THEN 3090
3080 GOSUB 7600: REM Draw Grid
3085 GOTO 3100
3090 PRINT #3, "PU;SP1;LT;PA"; GX; ","; GY; ";": GOSUB 60000
3091 PRINT #3, "EA"; PX; ","; PY; ";": GOSUB 60000
3099 REM *** Set up Colour Shading ***
3100 IF CSEL$ = "S" THEN 3200
3110 NOCL = 13
3120 C$ = "1314051204101102030901060815"
3130 FOR K = 0 TO NOCL
3140 C(K) = VAL(MID$(C$, K * 2 + 1, 2))
3150 NEXT K
3160 TCC = (ZMAX - ZMIN) / (NOCL + 1)
3170 GOSUB 7100: CI = TCC
3180 HC = INT(ZMAX / TCC) * TCC: LC = INT(ZMIN / TCC) * TCC
3190 NOCL = (HC - LC) / TCC: GOSUB 7800
3200 IF CSEL$ = "A" THEN 3300
3230 CLS: GOSUB 9000
3240 LOCATE 8, 10: PRINT "CONTOUR LIMITS"
3250 LOCATE 10, 10: INPUT "LOW VALUE "; LC
3252 IF LC < ZMIN THEN 3250
3255 IF LC > ZMAX THEN 3250
3260 LOCATE 12, 10: INPUT "HIGH VALUE "; HC
3262 IF HC > ZMAX THEN 3260
3265 IF HC < ZMIN THEN 3260
3268 IF HC < LC THEN 3260
3270 LOCATE 14, 10: INPUT "CONTOUR INTERVAL "; CI
3276 IF CI <= 0 THEN 3270
3300 FOR CC = LC TO HC STEP CI
```

```
3302 LOCATE 16, 10: PRINT "CURRENT CONTOUR = "; CC
3305 PRINT #3, "SI0.18,0.22;": GOSUB 60000
3309 CT = CC / 10
3310 IF CT = INT(CT) THEN PRINT #3, "PU;SP3;": GOTO 3320
3311 CT = (CC - 5) / 10
3312 IF CT = INT(CT) THEN PRINT #3, "PU;SP4;": GOTO 3320
3313 CT = (CC - 2.5) / 5
3314 IF CT = INT(CT) THEN PRINT #3, "PU;SP2;": GOTO 3320
3315 CT = (CC - .5)
3316 IF CT = INT(CT) THEN PRINT #3, "PU;SP3;": GOTO 3320
3317 CT = (CC - .25) * 10
3318 IF CT = INT(CT) THEN PRINT #3, "PU;SP4;": GOTO 3320
3319 PRINT #3, "PU;SP1;"
3320 FOR I = 0 TO (NROW - 2)
3330 Y0 = MG * I
3340 FOR J = 0 TO (NCOL - 2)
3350 X0 = MG * J
3360 NP = 0: CLOR = INT((HC - CC) / TCC)
3370 Z1 = Z(I, J)
3380 IF CC > Z1 THEN NP = NP + 1
3390 Z2 = Z(I, (J + 1))
3400 IF CC > Z2 THEN NP = NP + 1
3410 Z3 = Z((I + 1), J)
3420 IF CC > Z3 THEN NP = NP + 1
3430 Z4 = Z((I + 1), (J + 1))
3440 IF CC > Z4 THEN NP = NP + 1
3450 IF NP = 0 OR NP = 4 GOTO 3840
3460 A = Z1
3470 B = Z2 - A
3480 C = Z3 - A
3490 D = Z4 - A - B - C
3500 ZT = 0
3510 FOR Y1 = 0 TO 1 STEP .25
3520 DR = B + D * Y1
3530 IF DR = 0 GOTO 3680
3540 X1 = (CC - A - C * Y1) / DR
3550 IF X1 < 0 OR X1 > 1 GOTO 3680
3560 X = X0 + MG * X1
3570 Y = Y0 + MG * Y1
3580 IF X > PX OR Y > PY GOTO 3670
3590 IF ZT = 0 THEN 3615
3600 PRINT #3, "PD;PA"; (X * SM + GX); ","; (Y + GY); ";"
3610 GOTO 3670
3615 PRINT #3, "PU;PA"; (X * SM + GX); ","; (Y + GY); ";"
3616 GOSUB 60000
3630 IF TL > 0 THEN 3670
3631 REM IF CC / 5 <> INT(CC / 5) THEN 3670
3632 LDAT = CC
3633 AVGX = (X * SM + GX)
```

```
3634 AVGY = (Y + GY)
3640 PRINT #3, "PU;PA"; AVGX; ","; AVGY; ";": GOSUB 60000
3645 PRINT #3, "LB"; LDAT; CHR$(3)
3650 TL = TL + 1
3660 GOTO 3615
3670 ZT = ZT + 1
3680 NEXT Y1
3690 ZT = 0
3700 FOR X1 = 0 TO 1 STEP .25
3710 DS = C + D * X1
3720 IF DS = 0 GOTO 3830
3730 Y1 = (CC - A - B * X1) / DS
3740 IF Y1 < 0 OR Y1 > 1 GOTO 3830
3750 X = X0 + MG * X1
3760 Y = Y0 + MG * Y1
3770 IF X > PX OR Y > PY GOTO 3820
3780 IF ZT = 0 THEN 3810
3790 PRINT #3, "PD"; (X * SM + GX); ","; (Y + GY); ";"
3800 GOTO 3820
3810 PRINT #3, "PU;PA"; (X * SM + GX) ; ","; (Y + GY); ";"
3811 GOSUB 60000
3820 ZT = ZT + 1
3830 NEXT X1
3840 NEXT J
3850 NEXT I
3860 TL = 0
3870 NEXT CC
3880 CLS: GOSUB 9000
3900 IF CSEL$ = "A" THEN 9530
3910 LOCATE 10, 10: PRINT "INPUT (M)ORE for"
3920 LOCATE 12, 10: PRINT "CONTOURS added"
3930 LOCATE 14, 10: PRINT "'(N)ew for Fresh Plot,"
3940 LOCATE 16, 10: PRINT "or (Q)uit"
3945 BEEP
3955 LOCATE 16, 25: PRINT "?"; : M$ = INPUT$(1)
3960 IF INSTR("MNQmnq", M$) = 0 THEN 3945
3965 IF M$ = "m" THEN M$ = "M"
3970 IF M$ = "n" THEN M$ = "N"
3975 IF M$ = "q" THEN M$ = "Q"
4010 IF M$ = "M" THEN 3200
4020 IF M$ = "N" THEN CLOSE: RUN MDSK$ + "GPLOT-D2"
4030 IF M$ = "Q" THEN CLOSE: GOTO 4040
4040 M$ = " ": SR% = 1
4050 GOTO 9530
```

APPENDIX D.3:
SHADED CONTOURS COMPUTER PROGRAM “GRAPH-D3”

This program was discussed in Section 4.3.2 and is suitable for drawing contours at regular intervals between the highest and lowest contours. Note, this program will only run on colour monitors with EGA or VGA graphics cards.

Shaded contours

(a) Initialization and control Line numbers 10–90

See section (a) for Appendix D.2.

(b) Data loading Line numbers 100–160

See section (b) for Appendix D.2.

(c) Shaded contour set-up Line numbers 200–460

This segment of code is similar to section (c) of Appendix D.2.

(d) Shaded contour display Line numbers 3000–3880

This is the main part of the program. The screen is set up as for other presentation graphics programs using three **GOSUB** routines at Lines 7300, 7500 and 7600. The colour shading is set up in Line 3260 using the string **C$,** and can be changed by the user if desired. Before the contour map is drawn, a colour shading panel is displayed in the right-hand side of the screen, giving the colours for each contour level, using the **PAINT** command.
This program also requires the routines (e) to (q) as listed for Appendix D.2.

“GRAPH-D3”—BASIC program

```
10 REM <GRAPH-D3> Presentation Graphics Program D.3
14 REM (C) Copyright P.H.Milne 1990 - Shaded Contours
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: **** Maximum 50,50 ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
65 IF SCR% <> 2 THEN 9600
70 GX0 = 70: GY0 = 50: A$ = "*": DATBX = 0: SR% = 2: SM = 2.4
80 DIM Z(50, 50), ZG(4, 4), C(15)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Shaded Contours"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
```

```
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
200 CLS : GOSUB 9000
210 LOCATE 6, 10: PRINT "Contour DATA File "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
340 XTC = (XMAX - XMIN) / 10
360 XM = 500
370 YTC = (YMAX - YMIN) / 10
390 YM = 120
400 LOCATE 20, 10: PRINT "Background Grid (Y/N)"; : A$ = INPUT$(1)
410 IF INSTR("YNyn", A$) = 0 THEN 400
420 IF A$ = "y" THEN A$ = "Y"
430 IF A$ = "n" THEN A$ = "N"
440 IF A$ = "Y" THEN HGRID = 1: VGRID = 1: YLAB$ = "V"
450 IF A$ = "N" THEN HGRID = 0: VGRID = 0: YLAB$ = "V"
460 GOSUB 8500
2995 REM ***** Screen Plot *****
3000 GRAPH = 2: SX = 240 * SR%: SY = 120
3010 GX = GX0: GY = GY0: MG = 0: MH = 0: MV = 0: PP = 0
3020 MH = SX / ((NCOL - 1) * SM): MV = SY / (NROW - 1)
3030 IF MH <= MV THEN MG = MH ELSE MG = MV
3050 IF SCR% = 1 THEN RGB% = 2: CL% = 1
3051 IF SCR% = 1 AND SD% = 2 THEN RGB% = 3: CL% = 1
3052 IF SCR% = 2 THEN RGB% = 15: CL% = 14
3053 IF SCR% = 3 THEN RGB% = 3: CL% = 1
3054 IF SCR% = 4 THEN RGB% = 3: CL% = 1
3060 PX = GX + (MG * (NCOL - 1) * SM): PY = GY + MG * (NROW - 1)
3070 GOSUB 7300: REM Set up Screen
3080 IF HGRID = 0 THEN 3100
3090 GOSUB 7600: REM Draw Grid
3100 LINE (GX, GY) - (PX, PY), RGB%, B
3110 GOSUB 7500
3195 IF PP = 1 THEN 3880
3200 REM *** Set up Colour Shading ***
3250 NOCL = 13
3260 C$ = "1314051204101102030901060815"
3270 FOR K = 0 TO NOCL
3280 C(K) = VAL(MID$(C$, K * 2 + 1, 2))
3290 NEXT K
3300 TCC = (ZMAX - ZMIN) / (NOCL + 1)
3305 GOSUB 7100
3310 HC = INT(ZMAX / TCC) * TCC: LC = INT(ZMIN / TCC) * TCC
3315 NOCL = (HC - LC) / TCC: GOSUB 7800
```

```
3320 FOR I = 0 TO (NROW - 2)
3330 Y0 = MG * I
3340 FOR J = 0 TO (NCOL - 2)
3350 X0 = MG * J
3370 ZG(0, 0) = Z(I, J)
3390 ZG(0, 4) = Z(I, (J + 1))
3410 ZG(4, 0) = Z((I + 1), J)
3430 ZG(4, 4) = Z((I + 1), (J + 1))
3500 FOR ROW0 = 1 TO 3
3510 ZG(0, ROW0) = ZG(0, 0) + .25 * ROW0 * (ZG(0, 4) - ZG(0, 0))
3520 NEXT ROW0
3530 FOR ROW4 = 1 TO 3
3540 ZG(4, ROW4) = ZG(4, 0) + .25 * ROW4 * (ZG(4, 4) - ZG(4, 0))
3550 NEXT ROW4
3560 FOR ROW = 1 TO 3
3570 FOR COL = 0 TO 4
3580 ZG(ROW, COL) = ZG(0, COL) + .25 * ROW * (ZG(4, COL) - ZG(0,
   COL))
3590 NEXT COL
3600 NEXT ROW
3610 FOR ROWZ = 0 TO 3
3620 FOR COLZ = 0 TO 3
3630 Z11 = ZG(ROWZ, COLZ)
3640 Z12 = ZG(ROWZ, (COLZ + 1))
3650 Z13 = ZG((ROWZ + 1), (COLZ + 1))
3660 Z14 = ZG((ROWZ + 1), COLZ)
3670 ZAV = (Z11 + Z12 + Z13 + Z14) / 4
3700 CLOR = INT((HC - ZAV) / TCC) + 1
3710 X1 = X0 + (.25 * COLZ) * MG
3720 Y1 = Y0 + (.25 * ROWZ) * MG
3730 X2 = X0 + .25 * (1 + COLZ) * MG
3740 Y2 = Y0 + .25 * (1 + ROWZ) * MG
3750 LINE (X1 * SM + GX, Y1 + GY) - (X2 * SM + GX, Y2 + GY) , C
    (CLOR) , BF
3770 NEXT COLZ
3780 NEXT ROWZ
3840 NEXT J
3850 NEXT I
3870 PP = 1: GOTO 3080
3880 GOTO 9500
```

Appendix E
3-D Charts and surface models

APPENDIX E.1:
3-D COLUMN CHARTS : 3-D AREA CHARTS

COMPUTER PROGRAM "GRAPH-E1"

This program was discussed in Section 5.2, and covers both 3-D column charts and 3-D area charts, as the two sets of routines are very similar.

3-D Column charts and 3-D Area charts

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables which require to be initialized for use in the program. As discussed in section (a) of Appendix C.1, the program has a maximum of five data sets with fifteen rows of data.

(b) Data loading Line numbers 100–160

See the notes for section (b) of Appendix B.1.

(c) Data check Line numbers 200–440?

See the notes for section (c) of Appendix B.1.

(d) Text file entry Line numbers 500–590

See the notes for section (d) of Appendix B.1.

(e) Graph presentation selection Line numbers 600–780

The user now has to choose how the graph is to be displayed, either as a 3-D column chart or as a 3-D area chart, with or without a background grid.

(f) Set up display screen Line numbers 1000–1090

This segment of code now sets up the screen **(GOSUB 7300),** titles the graph **(GOSUB 7500),** and draws a background grid if required **(GOSUB 7600)**.

(g) 3-D Column chart display Line numbers 1200–1360

This is the main part of the 3-D column chart presentation and draws the data series from back to front, as described in Section 5.2.1. If a colour screen is being used, each data set will be colour shaded accordingly using the **PAINT** command.

(h) 3-D Area chart display Line numbers 1500–1950

This is the main part of the 3-D area chart presentation, as described in Section 5.2.2. The method of display is the same as that described in (g) above for 3-D column charts.
This program also requires the following routines.

(i) Appendix A.8	Line numbers 7000–7090
(j) Appendix A.9	Line numbers 7100–7290
(k) Appendix A.10	Line numbers 7300–7490
(l) Appendix A.11	Line numbers 7500–7590
(m) Appendix A.12	Line numbers 7600–7710
(n) Appendix A.2	Line numbers 8000–8180
(o) Appendix A.3	Line numbers 8200–8390
(p) Appendix A.4	Line numbers 8400–8490
(q) Appendix A.5	Line numbers 8500–8590
(r) Appendix A.3	Line numbers 8800–8990
(s) Appendix A.15	Line numbers 9000–9050
(t) Appendix A.16	Line numbers 9500–9600
(u) Appendix A.17	Line numbers 10000–10310
(v) Appendix A.18	Line numbers 40000–40590
(w) Appendix A.19	Line numbers 55000–56960

“GRAPH-E1”—BASIC program

```
10 REM <GRAPH-E1> Presentation Graphics Program E.1 - 3D Graphs
14 REM (C) Copyright P.H.Milne 1990
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: **** Max 15 Rows of Data ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 X0 = 70: Y0 = 50: A$ = "*": DATBX = 0
70 DIM Z(15, 6), CL%(10)
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
```

```
100 PC$ = "3D Graph"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of DATA Sets = "; ND
130 LOCATE 17, 10: PRINT "Number of DATA Points = "; NROW
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9500
200 CLS: GOSUB 9000
210 GOSUB 7000: REM Check Min & Max Data
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 15, 10: PRINT "Change Default Values (Y/N) ";
265 A$ = INPUT$(1)
270 IF INSTR("YNyn", A$) = 0 THEN 260
280 IF YMIN > 0 THEN YMIN = 0
290 IF YMIN < 0 THEN YMIN = YMIN - 10
300 IF A$ = "N" OR A$ = "n" THEN 340
320 LOCATE 18, 10: INPUT "New Y-AXIS (Y-max) "; YMAX
330 IF YMIN >= YMAX THEN 320
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 360
370 YTC = (YMAX - YMIN) / 5
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 60
400 GOSUB 8500: REM Read Titles from Disc
410 LOCATE 20, 10: PRINT "Change Graph Titles (Y/N) ";
420 A$ = INPUT$(1)
430 IF INSTR("YNyn", A$) = 0 THEN 410
440 IF A$ = "N" OR A$ = "n" THEN 600
499 REM ***** JUMPS HERE FROM ERROR IN 8500 *****
500 CLS: GOSUB 9000
510 LOCATE 6, 10: INPUT "Enter Graph Title (Max 40) "; TITLE$
520 LOCATE 8, 10: INPUT "Enter X-AXIS Title (Max 30) "; LXAXI$
530 LOCATE 10, 10: INPUT "Enter Y-AXIS Title (Max 11) "; LYAXI$
540 YLAB$ = "V"
550 FOR I = 1 TO NCOL - 1
560 LOCATE 12 + I, 10: PRINT "Enter Legend DATA "; I; " (Max 9) ";
570 INPUT DAT$(I)
580 NEXT I
590 GOSUB 8400: REM Store Titles on Disc
600 CLS: GOSUB 9000
610 LOCATE 5, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 8, 10: PRINT "<1> 3D Column Chart with Background Grid"
630 LOCATE 10, 10: PRINT "<2> 3D Column Chart only"
640 LOCATE 12, 10: PRINT "<3> 3D Area Chart with Background Grid"
```

```
650 LOCATE 14, 10: PRINT "<4> 3D Area Chart only"
670 LOCATE 20, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("1234", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720, 730, 740
710 HGRID = 1: DATBX = 1: YLAB$ = "V": GOTO 750
720 HGRID = 0: DATBX = 1: YLAB$ = "V": GOTO 750
730 HGRID = 1: DATBX = 1: YLAB$ = "V": GOTO 750
740 HGRID = 0: DATBX = 1: YLAB$ = "V"
750 CLS: GOSUB 9000
760 X3ND = 120 / (2 * 5): Y3ND = 60 / (2 * 5): REM Maximum ND = 5
770 XCND = INT(360 / (2 * NROW))
780 X0F = X0 + 10 + XCND / 2
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500: REM Titles
1020 FOR J = 1 TO NCOL - 1
1030 AVGX = 94 + 103 * (J - 1): AVGY = Y0 - 22
1040 GOSUB 40000
1050 LOCATE 22, 1 + J * 13: PRINT DAT$(J)
1060 NEXT J
1070 IF HGRID = 0 THEN 1090
1080 GOSUB 7600: REM Draw Grid
1090 ON NSEL GOTO 1200, 1200, 1500, 1500
1199 REM **** 3D Column Chart ****
1200 FOR J = NCOL TO 2 STEP -1
1210 FOR K = 1 TO NROW
1220 X1 = (K - 1) * (2 * XCND) + X0 + 10 + (2 * J - 3) * X3ND
1230 X2 = X1 + XCND: Y1 = Y0 + (2 * J - 3) * Y3ND
1240 Y2 = (Z(K, J) - YMIN) * 12 / YTC + Y0 + (2 * J - 3) * Y3ND
1250 LINE (X1, Y1) - (X2, Y2), CL%(J + 4), B
1255 IF SCR% <> 2 THEN 1270
1260 PAINT (X1 + XCND / 2, Y1 + 2), CL%(J - 1), CL%(J + 4)
1270 LINE (X1, Y2) - (X1 + X3ND, Y2 + Y3ND), CL%(J + 4)
1280 LINE - (X2 + X3ND, Y2 + Y3ND), CL%(J + 4)
1290 LINE - (X2, Y2), CL%(J + 4)
1295 IF SCR% <> 2 THEN 1310
1300 PAINT (X1 + XCND, Y2 + 2), CL%(J - 1), CL%(J + 4)
1310 LINE (X2 + X3ND, Y2 + Y3ND) - (X2 + X3ND, Y1 + Y3ND), CL%(J +
   4)
1320 LINE - (X2, Y1), CL%(J + 4)
1325 IF SCR% <> 2 THEN 1340
1330 PAINT (X2 + 2, Y2), CL%(J - 1), CL%(J + 4)
1340 NEXT K
1350 NEXT J
1360 GOTO 9500
1499 REM ***** Area Graph *****
1500 FOR J = NCOL TO 2 STEP -1
1510 X1 = (Z(1, 1) - XMIN) * 36 / XTC + X0F + (2 * J - 3) * X3ND
1520 X2 = X1: Y1 = Y0 + (2 * J - 3) * Y3ND
```

```
1530 Y2 = (Z(1, J) - YMIN) * 12 / YTC + Y0 + (2 * J - 3) * Y3ND
1540 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1550 FOR K = 1 TO NROW - 1
1560 X1 = (K - 1) * (2 * XCND) + X0F + (2 * J - 3) * X3ND
1570 Y1 = (Z(K, J) - YMIN) * 12 / YTC + Y0 + (2 * J - 3) * Y3ND
1580 X2 = (K + 1 - 1) * (2 * XCND) + X0F + (2 * J - 3) * X3ND
1590 Y2 = (Z(K + 1, J) - YMIN) * 12 / YTC + Y0 + ( 2 * J - 3) * Y3ND
1600 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1610 NEXT K
1620 X1 = X2: Y1 = Y2
1640 X2 = X1: Y2 = Y0 + (2 * J - 3) * Y3ND
1650 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1660 X1 = X2: Y1 = Y2
1670 X2 = (Z(1, 1) - XMIN) * 36 / XTC + X0F + (2 * J - 3) * X3ND
1680 Y2 = Y1
1690 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1695 IF SCR% <> 2 THEN 1710
1700 PAINT (X2 + 10, Y2 + 2), CL%(J - 1), CL%(J + 4)
1710 FOR K = 1 TO NROW - 1
1720 X1 = (K - 1) * (2 * XCND) + X0F + (2 * J - 3) * X3ND
1730 Y1 = (Z(K, J) - YMIN) * 12 / YTC + Y0 + (2 * J - 3) * Y3ND
1740 X2 = (K + 1 - 1) * (2 * XCND) + X0F + (2 * J - 3) * X3ND
1750 Y2 = (Z(K + 1, J) - YMIN) * 12 / YTC + Y0 + (2 * J - 3) * Y3ND
1760 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1770 LINE - (X2 + X3ND, Y2 + Y3ND) , CL%(J + 4)
1780 LINE - (X1 + X3ND, Y1 + Y3ND), CL%(J + 4)
1790 LINE - (X1, Y1), CL%(J + 4)
1800 IF SCR% <> 2 THEN 1830
1810 PAINT (X1 + X3ND / 2 + 2, Y1 + Y3ND/2), CL%(J - 1), CL%(J + 4)
1830 NEXT K
1840 X1 = X2: Y1 = Y2
1850 X2 = X1: Y2 = Y0 + (2 * J - 3) * Y3ND
1860 LINE (X1, Y1) - (X2, Y2), CL%(J + 4)
1870 LINE - (X2 + X3ND, Y2 + Y3ND), CL%(J + 4)
1880 LINE - (X1 + X3ND, Y1 + Y3ND), CL%(J + 4)
1890 IF SCR% <> 2 THEN 1920
1900 PAINT (X1 + 2, Y1), CL%(J - 1), CL%(J + 4)
1920 NEXT J
1950 GOTO 9500
```

APPENDIX E.2:

LINE PROFILES COMPUTER PROGRAM “GRAPH-E2”

This program was discussed in Section 5.3.1 and displays topographic data **(‘.CON’)** as a line profile.

Line profiles

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice, and the numeric and string variables which require to be initialized for use in the program. For memory considerations, note that the maximum number of grid points, i.e. rows and columns is 50 each, **Z(50,50)**.

(b) Data loading — Line numbers 100–160

See the notes for section (b) of Appendix D.2.

(c) Digital ground model data — Line numbers 200–390

This segment of code displays the maximum and minimum *X, Y* and *Z* values in the data file.

(d) Graph presentation selection — Line numbers 600–790

The user now has to choose whether to display the line profiles with or without a background grid.

(e) Set up display screen — Line numbers 1000–1080

This segment of code now sets up the screen **(GOSUB 7300),** titles the graph **(GOSUB 7500)** and draws a background grid if required **(GOSUB 7600)**.

(f) Line profile display — Line numbers 1500–1950

This segment of code is similar to the 3-D area chart display (Appendix E.1), but without depth enhancement to allow up to 50 rows to be displayed.

This program also requires routines (i)–(w) with the exception of (p) and (v) from Appendix E.1. Note that routine (1), Appendix A.11, requires an additional Line **'7531 RETURN'** as there is no *Y*-axis label.

"GRAPH-E2"—BASIC program

```
10 REM <GRAPH-E2> Presentation Graphics Appendix E.2-3D Graphs
14 REM (C) Copyright P.H.Milne 1990 *** - Line Profile ***
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: *** Maximum 50,50 of Data ***
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
70 GX0 = 70: GY0 = 50: A$ = "*": DATBX = 0
80 DIM Z(50, 50), CL%(10)
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "Line Profile"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
145 A$ = INPUT$(1)
```

```
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
200 CLS: GOSUB 9000
210 LOCATE 6, 10: PRINT "Digital Ground Model "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
300 ZMIN = INT(ZMIN / 10) * 10: ZMAX = (INT(ZMAX / 10) + 1) * 10
340 XTC = (XMAX - XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 360
370 YTC = (ZMAX - ZMIN) / 5
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 60
600 REM Continue on same screen
610 LOCATE 16, 10: PRINT "Select Graph Presentation :-"
620 LOCATE 18, 10: PRINT "<1> 3D Profile with Background Grid"
630 LOCATE 20, 10: PRINT "<2> 3D Profile only"
670 LOCATE 22, 10: PRINT "Enter Choice "; : SEL$ = INPUT$(1)
680 IF INSTR("12", SEL$) = 0 THEN 670
690 NSEL = VAL(SEL$)
700 ON NSEL GOTO 710, 720
710 HGRID = 1: DATBX = 1: GOTO 750
720 HGRID = 0: DATBX = 1: GOTO 750
750 CLS: GOSUB 9000
760 X3ND = 120 / (2 * NROW): Y3ND = 60 / (2 * NROW)
770 XCND = (360/(2 * NCOL))
780 X0F = GX0 + 10 + XCND / 2
790 GOSUB 8500
1000 GRAPH = 2: GOSUB 7300: REM Sets up Screen
1010 GOSUB 7500
1070 IF HGRID = 0 THEN 1090
1080 GOSUB 7600: REM Draw Grid
1090 CL%(1) = 4: CL%(2) = 15
1499 REM ***** Line Profile *****
1500 FOR J = NROW - 1 TO 0 STEP -1
1505 FOR CLOR = 1 TO 2
1510 X1 = (XMIN) * 36 / XTC + X0F + (2 * J) * X3ND
1520 X2 = X1: Y1 = GY0 + (2 * J) * Y3ND
1530 Y2 = (Z(J, 0) - ZMIN) * 12 / YTC + GY0 + (2 * J) * Y3ND
1540 LINE (X1, Y1) - (X2, Y2), CL%(CLOR)
1550 FOR K = 0 TO NCOL - 2
1560 X1 = (K) * (2 * XCND) + X0F + (2 * J) * X3ND
1570 Y1 = (Z(J, K) - ZMIN) * 12 / YTC + GY0 + (2 * J) * Y3ND
1580 X2 = (K + 1) * (2 * XCND) + X0F + (2 * J) * X3ND
1590 Y2 = (Z(J, K + 1) - ZMIN) * 12 / YTC + GY0 + (2 * J) * Y3ND
```

```
1600 LINE (X1, Y1) - (X2, Y2), CL%(CLOR)
1610 NEXT K
1620 X1 = X2: Y1 = Y2
1640 X2 = X1: Y2 = GY0 + (2 * J) * Y3ND
1650 LINE (X1, Y1) – (X2, Y2), CL%(CLOR)
1660 X1 = X2: Y1 = Y2
1670 X2 = (XMIN) * 36 / XTC + X0F + (2 * J) * X3ND
1680 Y2 = Y1
1690 LINE (X1, Y1) – (X2, Y2), CL%(CLOR)
1695 IF SCR% <> 2 THEN 1920
1696 IF CLOR = 2 THEN 1920
1700 PAINT (X2 + 10, Y2 + 1), 0, CL%(CLOR)
1710 NEXT CLOR
1920 NEXT J
1950 GOTO 9500
```

APPENDIX E.3:
3-D LINE PROFILES COMPUTER PROGRAM "GRAPH-E3"

This program was discussed in Section 5.3.2 and displays topographic data **('.CON')** or surface data **('.EQD')** as a 3-D line profile.

3-D line profiles

(a) Initialization and control Line numbers 10–90

See the notes for section (a) of Appendix E.2.

(b) Data loading Line numbers 100–180

See the notes for section (b) of Appendix D.2.

(c) Digital ground model data Line numbers 200–580

This segment of code displays the maximum and minimum *X, Y* and *Z* values in the data file. If a colour screen is being used, the string **C$** sets up the colour shading for the line profiles.

(d) 3-D line profile display Line numbers 4000–4330

This is the main part of the program, and sets up an isometric projection of the digital ground model. If a colour screen is being used, a colour shaded panel is displayed on the right-hand side of the screen to identify the colour shaded levels. This program also requires the routines (j)–(u), with the exception of (k), (m) and (p), as listed for Appendix E.1. An additional routine, listed in Appendix A.13, is also required at Line 7800.

"GRAPH-E3"—BASIC program

```
10 REM <GRAPH-E3> Presentation Graphics Appendix E.3 - 3D Graphs
14 REM (C) Copyright P.H.Milne 1990 - 3D Line Profile
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: **** Maximum 50,50 Data ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 DIM Z(50, 50), CL%(15)
70 GX0 = 70: GY0 = 50: A$ = "*": DATBX = 0
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 CLS: PC$ = "3D Profile"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
170 SR% = 2: SRX% = 2: SRY% = 1
180 IF SCR% = 2 THEN SRH% = 2 ELSE SRH% = 1
200 CLS: GOSUB 9000
210 LOCATE 6, 10: PRINT "Digital Ground Model - "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
400 LOCATE 20, 10: PRINT "Display 3D Profile (Y/N) ";
410 A$ = INPUT$(1)
420 IF INSTR("YNyn", A$) = 0 THEN 400
430 IF A$ = "N" OR A$ = "n" THEN 9600
440 GOSUB 8500: CLS
500 NOCL = 13: RGB% = 15
510 C$ = "1314051204101102030901060815"
520 FOR I = 0 TO NOCL
530 CL%(I) = VAL(MID$(C$, 1 * 2 + 1, 2))
540 NEXT I
550 TCC = (ZMAX - ZMIN) / 14
560 GOSUB 7100
570 HC = INT(ZMAX / TCC) * TCC: LC = INT(ZMIN / TCC) * TCC
580 NOCL = (HC - LC) / TCC
3995 REM ***** Screen Plotting Routine *****
4000 DISP% = 1: YCEN = 0: XCEN = 0: SX = 160 * SRX%: SY = 100
4003 IF SCR% <> 2 THEN RGB% = 2 ELSE RGB% = 12
4005 IF SCR% <> 2 THEN 4010
4007 SCREEN 9: COLOR 15, 0
4008 GX = GX0: GY = GY0
```

```
4010 VIEW SCREEN (5, 30) - (310 * SRX%, 150 * SRH%), 0, RGB%
4015 CLS : GOSUB 9000: IF SCR% <> 2 THEN 4025
4018 IF SCR% = 2 THEN GOSUB 7800
4020 VIEW SCREEN (5, 30) - (258 * SRX%, 150 * SRH%), 0, RGB%
4025 WINDOW (XCEN - SX, YCEN - SY) - (XCEN + SX, YCEN + SY)
4030 VEX = 50 / (ZMAX - ZMIN): YSMIN = INT((ZMIN * VEX) / 10) * 10
4035 IF SCR% <> 2 THEN 4045
4040 X0 = -30 * SRX%: GOTO 4050
4045 X0 = -10 * SRX%
4050 Y0 = (70 - YSMIN) * SRY%: GOSUB 7500
4070 S = 2: B = 2.5 * SRX%
4090 FOR GCOL = 1 TO NCOL - 1
4100 FOR GROW = 1 TO NROW - 1
4110 X1 = X0 + (GCOL - GROW) * B
4115 GRC1 = (GROW + GCOL) * S
4120 Y1 = Y0 + (Z(GROW, GCOL) * VEX - GRC1) * SRY%
4130 X2 = X1 - B
4135 GRC2 = (GROW + GCOL - 1) * S
4140 Y2 = Y0 + (Z(GROW, GCOL - 1) * VEX - GRC2) * SRY%
4150 X3 = X1 + B
4155 GRC3 = (GROW - 1 + GCOL) * S
4160 Y3 = Y0 +( Z(GROW - 1, GCOL) * VEX - GRC3) * SRY%
4170 X4 = X1
4175GRC4 = (GROW - 1 + GCOL - 1) * S
4180 Y4 = Y0 + (Z(GROW - 1, GCOL - 1) * VEX - GRC4) * SRY%
4190 IF SCR% <> 2 THEN CLOR = 13: GOTO 4250
4200 ZAV = (Z(GROW, GCOL - 1) + Z(GROW - 1, GCOL - 1)) / 2
4210 CLOR = INT((HC - ZAV) / TCC)
4220 IF CLOR < 0 THEN CLOR = 0
4250 LINE (X4, Y4) - (X2, Y2), CL%(CLOR)
4260 IF GCOL = NCOL - 1 THEN LINE (X3, Y3) - (X1, Y1), CL%(CLOR)
4270 NEXT GROW, GCOL
4271 NROW = NROW - 1: NCOL = NCOL - 1
4275 X1 = X0 + (-NROW) * B
4276 Y1 = Y0 + ((-NROW) * S + Z(NROW, 0) * VEX) * SRY%
4277 X2 = X0 + (-NROW) * B
4278 Y2 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4280 LINE (X1, Y1) - (X2, Y2), 15
4285 X1 = X0 + (NCOL - NROW) * B
4286 Y1 = Y0 + (-(NCOL + NROW) * S + Z(NROW, NCOL) * VEX) * SRY%
4287 X2 = X0 + (NCOL - NROW) * B
4288 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4290 LINE (X1, Y1) - (X2, Y2), 15
4295 X1 = X0 + (NCOL) * B
4296 Y1 = Y0 + ((-NCOL) * S + Z(0, NCOL) * VEX) * SRY%
4297 X2 = X0 + (NCOL) * B
4298 Y2 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4300 LINE (X1, Y1) - (X2, Y2), 15
4305 X1 = X0 + (-NROW) * B
```

```
4306 Y1 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4307 X2 = X0 + (NCOL - NROW) * B
4308 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4310 LINE (X1, Y1) - (X2, Y2), 15
4317 X2 = X0 + (NCOL) * B
4318 Y2 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4320 LINE -(X2, Y2), 15
4330 GOTO 9500
```

COMPUTER PROGRAM "GPLOT-E3"

To convert the program **"GRAPH-E3"** from screen graphics to an HPGL plotter the following changes are required:

replace segment (d) with the code for Lines 4000–4370, which are given at the end of this section
replace segment (k) with Appendix A.28
replace segment (1) with Appendix A.29
delete segments (v) and (w)
add the following:

(x) Appendix A.19	Line numbers 50000–50180
(y) Appendix A.20	Line numbers 50500–50570
(z) Appendix A.24	Line numbers 60000–60150

"GPLOT-E3"—BASIC program—segment (d) only

```
4000 DISP% = 1: YCEN = 0: XCEN = 0
4001 SX = 160 * SRX%: SY = 100
4010 GX = XCEN - SX: GY = YCEN - SY
4015 PX = XCEN + SX: PY = YCEN + SY
4020 PGX = GX - 10: PGY = GY - 20
4025 PPX = PX + 20: PPY = PY + 20
4030 VEX = 50 / (ZMAX - ZMIN)
4035 YSMIN = INT((ZMIN * VEX) / 10) * 10 + 5
4040 X0 = -10 * SRX%: Y0 = (70 - YSMIN) * SRY%
4050 CLS: YLAB$ = "V"
4060 GOSUB 9000: GOSUB 7300
4065 GOSUB 50500: GOSUB 7500
4070 S = 2: B = 2.5 * SRX%
4080 PRINT #3, "PU;SP3;": GOSUB 60000
4090 FOR GCOL = 0 TO NCOL - 1: TSEC = 1: GOSUB 60000
4095 PRINT #3, "PU;": GOSUB 60000: TSEC = 0
4100 FOR GROW = 0 TO NROW - 2
4110 X1 = X0 + (GCOL - GROW) * B
4115 YY = (Z(GROW, GCOL) * VEX - (GROW + GCOL) * S) * SRY%
4120 Y1 = Y0 + YY: Y1 = INT(Y1 * 10 + .5) / 10
```

```
4121 IF GCOL > 0 THEN 4130
4122 IF GROW > 0 THEN 4130
4123 IF Y1 > PY THEN Y0 = PY - YY: Y1 = Y0 + YY
4130 X2 = X1 - B
4135 GRC2 = (GROW + 1 + GCOL) * S
4140 Y2 = Y0 + (Z(GROW + 1, GCOL) * VEX - GRC2) * SRY%
4145 Y2 = INT(Y2 * 10 + .5) / 10
4150 PRINT #3, "PA"; X1; ","; Y1; ";"
4160 PRINT #3, "PD;PA"; X2; ","; Y2; ","
4170 NEXT GROW
4171 GOSUB 60000
4172 NEXT GCOL
4275 NROW = NROW - 1: NCOL = NCOL - 1: TSEC = 1
4280 BX1 = X0 + (-NROW) * B
4281 BTY1 = Y0 + ((-NROW) * S + Z(NROW, 0) * VEX) * SRY%
4285 BBY1 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4290 BX2 = X0 + (NCOL - NROW) * B
4291 BTY2 = Y0 + (-(NCOL + NROW) * S + Z(NROW, NCOL) * VEX) * SRY%
4295 BBY2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4300 BX3 = X0 + (NCOL) * B
4301 BTY3 = Y0 + ((-NCOL) * S + Z(0, NCOL) * VEX) * SRY%
4305 BBY3 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4310 PRINT #3, "PU;SP4;"
4315 PRINT #3, "PU;PA"; BX1; ","; BTY1; ";"
4320 PRINT #3, "PD;PA"; BX1; ","; BBY1; ";"
4325 PRINT #3, "PA";. BX2; ","; BBY2; ";"
4330 PRINT #3, "PA"; BX3; ","; BBY3; ";"
4335 PRINT #3, "PA"; BX3; ","; BTY3; ";PU;"
4340 PRINT #3, "PA"; BX2; ","; BTY2; ";"
4350 PRINT #3, "PD;PA"; BX2; ","; BBY2; ";PU;"
4360 PRINT #3, "PU;PA"; PGX; ","; PGY; ";SP0;": CLOSE #3
4370 GOTO 9530
```

APPENDIX E.4:

3-D OPEN MESH COMPUTER PROGRAM "GRAPH-E4"

This program was described in Section 5.3.3 and will display a 3-D fishnet mesh diagram for either topographic **('.CON')** or surface **('.EQD')** data files. Readers of my previous book *Computer Graphics for Surveying* will note that this program is an extension of **"PC-SURFPLOT"**.

3-D open mesh

(a) Initialization and control Line numbers 10–90

See the notes for section (a) of Appendix E.2.

(b) Data loading Line numbers 100–180

See the notes for section (b) of Appendix D.2.

(c) Digital ground model data

Line numbers 200–440

See the notes for section (c) of Appendix E.2.

(d) 3-D open mesh display

Line numbers 4000–4330

This is the main part of the program and will display an open mesh diagram. The mesh is drawn from back to front so that any old plotting underneath a new mesh segment is removed using the **PAINT** command.
This program also requires the routines (l)–(u), with the exception of (m) and (p) as listed for Appendix E.1.

"GRAPH-E4"—BASIC program

```
10 REM <GRAPH-E4> Presentation Graphics Appendix E.4 - 3D Graphs
14 REM (C) Copyright P.H.Milne 1990 - Open Mesh
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 23/05/90: **** Maximum 50,50 Data ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 DIM Z(50, 50), CL%(15)
70 GX0 = 70: GY0 = 50: A$ = "*": DATBX = 0
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "3D Surfplot"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Graph (Y/N) ";
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
170 SRX% = 2: SRY% = 1
180 IF SCR% = 2 THEN SRH% = 2 ELSE SRH% = 1
200 CLS: GOSUB 9000
210 LOCATE 6, 10: PRINT "Digital Ground Model - "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
400 LOCATE 20, 10: PRINT "Display Open Mesh (Y/N) ";
410 A$ = INPUT$(1)
420 IF INSTR("YNyn", A$) = 0 THEN 400
430 IF A$ = "N" OR A$ = "n" THEN 9600
440 GOSUB 8500: CLS
```

```
3995 REM ***** Screen Plotting Routine *****
4000 DISP% = 1: YCEN = 0: XCEN = 0
4001 SX = 160 * SRX%: SY = 100
4003 IF SCR% <> 2 THEN RGB% = 2 ELSE RGB% = 12
4005 IF SCR% <> 2 THEN 4010
4007 SCREEN 9: COLOR 15, 0
4010 VIEW SCREEN (5, 30) - (310 * SRX%, 150 * SRH%), 0, RGB%
4020 WINDOW (XCEN - SX, YCEN - SY) - (XCEN + SX, YCEN + SY)
4030 VEX = 50 / (ZMAX - ZMIN)
4035 YSMIN = INT((ZMIN * VEX) / 10) * 10 + 5
4040 X0 = -10 * SRX%: Y0 = (70 - YSMIN) * SRY%
4050 CLS : YLAB$ = "V"
4060 GOSUB 9000: GOSUB 7500
4070 S = 2: B = 2.5 * SRX%
4090 FOR GCOL = 1 TO NCOL - 1
4100 FOR GROW = 1 TO NROW - 1
4110 X1 = X0 + (GCOL - GROW) * B
4115 GRC1 = (GROW + GCOL) * S
4120 Y1 = Y0 + (Z(GROW, GCOL) * VEX - GRC1) * SRY%
4130 X2 = X1 - B
4135 GRC2 = (GROW + GCOL - 1) * S
4140 Y2 = Y0 + (Z(GROW, GCOL - 1) * VEX - GRC2) * SRY%
4150 X3 = X1 + B
4155 GRC3 = (GROW - 1 + GCOL) * S
4160 Y3 = Y0 + (Z(GROW - 1, GCOL) * VEX - GRC3) * SRY%
4170 X4 = X1
4175 GRC4 = (GROW - 1 + GCOL - 1) * S
4180 Y4 = Y0 + (Z(GROW - GCOL - 1) * VEX - GRC4) * SRY%
4190 LINE (X1, Y1) - (X2, Y2), 4
4200 LINE - (X4, Y4), 4: LINE - (X3, Y3), 4
4210 LINE - (X1, Y1), 4
4220 Y = (Y1 + Y4) / 2
4230 PAINT (X4, Y), 0, 4
4240 LINE (X1, Y1) - (X2, Y2), 15
4250 LINE - (X4, Y4), 15: LINE - (X3, Y3), 15
4260 LINE - (X1, Y1), 15
4270 NEXT GROW, GCOL
4271 NROW = NROW - 1: NCOL = NCOL - 1
4275 X1 = X0 + (-NROW) * B
4276 Y1 = Y0 + ((-NROW) * S + Z(NROW, 0) * VEX) * SRY%
4277 X2 = X0 + (-NROW) * B
4278 Y2 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4280 LINE (X1, Y1) - (X2, Y2), 15
4285 X1 = X0 + (NCOL - NROW) * B
4286 Y1 = Y0 + (-(NCOL + NROW) * S + Z(NROW, NCOL) * VEX) * SRY%
4287 X2 = X0 + (NCOL - NROW) * B
4288 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4290 LINE (X1, Y1) - (X2, Y2), 15
4295 X1 = X0 + (NCOL) * B
```

```
4296 Y1 = Y0 + ((-NCOL) * S + Z(0, NCOL) * VEX) * SRY%
4297 X2 = X0 + (NCOL) * B
4298 Y2 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4300 LINE (X1, Y1) - (X2, Y2), 15
4305 X1 = X0 + (-NROW) * B
4306 Y1 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4307 X2 = X0 + (NCOL - NROW) * B
4308 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4310 LINE (X1, Y1) - (X2, Y2), 15
4317 X2 = X0 + (NCOL) * B
4318 Y2 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4320 LINE - (X2, Y2), 15
4330 GOTO 9500
```

COMPUTER PROGRAM "GPLOT-E4"

To convert the program **"GRAPH-E4"** from screen graphics to an HPGL plotter take the previous plotter program **"GPLOT-E3"** and make the following changes:

replace "E3" with "E4" on Lines 10 and 9580
- replace "3D Profile" with "3D Surfplot" on Lines 14 and 100
- replace "3D Profile" with "Open Mesh" on Line 400
- add to segment (d) the code for Lines 4190–4272, which are given at the end of this section

"GPLOT-E4"—BASIC program—addition to segment (d) only

```
4190 FOR GROW - 0 TO NROW - 1: TSEC = 1: GOSUB 60000
4195 PRINT #3, "PU;": GOSUB 60000: TSEC = 0
4200 FOR GCOL = 0 TO NCOL - 2
4210 X1 = X0 + (GCOL - GROW) * B
4220 Y1 = Y0 + (Z(GROW, GCOL) * VEX - (GROW + GCOL) * S) * SRY%
4225 Y1 = INT(Y1 * 10 + .5) / 10
4230 X2 = X0 + (GCOL + 1 - GROW) * B
4235 GRC2 = (GROW + GCOL + 1) * S
4240 Y2 = Y0 + (Z(GROW, GCOL + 1) * VEX - GRC2) * SRY%
4245 Y2 = INT(Y2 * 10 + .5) / 10
4250 PRINT #3, "PA"; X1; ","; Y1; ";"
4260 PRINT #3, "PD;PA"; X2; ","; Y2; ";"
4270 NEXT GCOL
4271 GOSUB 60000
4272 NEXT GROW
```

APPENDIX E.5:

3-D SHADED MESH COMPUTER PROGRAM "GRAPH-E5"

This program was described in Section 5.3.4, and can be used to display either topographic **('.CON')** or surface **('.EQD')** data files. This program is an extension of the 3-D open mesh program to colour shade each mesh segment depending on its average height in the digital ground model.

3-D shaded mesh

See notes (a)–(d) for Appendix E.4. This program also requires routines (j)–(u), with the exception of (k), (m) and (p) in Appendix E.1. An additional routine, listed in Appendix A.13 is also required at Line 7800.

"GRAPH-E5"—BASIC program

```
10 REM <GRAPH-E5> Presentation Graphics Appendix E.5 - 3D Graphs
14 REM (C) Copyright P.H.Milne 1990 *** - Shaded Mesh ***
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 1990: **** Maximum 50,50 Data ****
30 CLEAR
40 ON ERROR GOTO 10000
50 PCF$ = "PGSCRDSK.PGD"
60 DIM 2(50, 50), CL%(15)
70 GX0 = 70: GY0 = 50: A$ = "*": DATBX = 0: SR% = 2
80 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 PC$ = "3D Surface"
110 GOSUB 8200: REM Check ID File and DATA
120 LOCATE 15, 10: PRINT "Number of Rows = "; NROW
130 LOCATE 17, 10: PRINT "Number of Columns = "; NCOL
140 LOCATE 20, 10: PRINT "Do you wish to VIEW Model (Y/N)"
145 A$ = INPUT$(1)
150 IF INSTR("YNyn", A$) = 0 THEN 140
160 IF A$ = "N" OR A$ = "n" THEN 9600
200 CLS: GOSUB 9000
210 LOCATE 6, 10: PRINT "Digital Ground Model - "; DFILEN$
220 LOCATE 10, 10: PRINT "X-AXIS : Min = "; XMIN
230 LOCATE 10, 40: PRINT ": Max = "; XMAX
240 LOCATE 12, 10: PRINT "Y-AXIS : Min = "; YMIN
250 LOCATE 12, 40: PRINT ": Max = "; YMAX
260 LOCATE 14, 10: PRINT "Z-AXIS : Min = "; ZMIN
270 LOCATE 14, 40: PRINT ": Max = "; ZMAX
340 XTC = (XMAX – XMIN) / 10
350 TCC = XTC: GOSUB 7100
360 XTC = TCC: XM = 360
370 YTC = (ZMAX – ZMIN) / 5
380 TCC = YTC: GOSUB 7100
390 YTC = TCC: YM = 60
400 LOCATE 20, 10: PRINT "Display 3D Surface (Y/N) ";
410 A$ = INPUT$(1)
```

```
420 IF INSTR("YNyn", A$) = 0 THEN 400
430 IF A$ = "N" OR A$ = "n" THEN 9600
440 GOSUB 8500: REM Read Text Title Data from Disc
3995 REM ***** Screen Plotting Routine *****
4000 GX = GX0: GY = GY0: YCEN = 0: XCEN = 0
4001 SX = 160 * SR%: SY = 100 * SR%
4005 SCREEN 9: COLOR 15, 0
4010 VIEW SCREEN (5, 30) - (310 * SR%, 150 * SR%), , 12
4050 CLS : GOSUB 9000: GOSUB 7500
4060 IF GSC = 1 THEN 4090
4063 NOCL = 13: RGB% = 15
4065 C$ = "1314051204101102030901060815"
4066 FOR I = 0 TO NOCL
4067 CL%(I) = VAL(MID$(C$, I * 2 + 1, 2))
4068 NEXT I
4070 TCC = (ZMAX - ZMIN) / 14
4075 GOSUB 7100
4078 HC = INT(ZMAX / TCC) * TCC: LC = INT(ZMIN / TCC) * TCC
4080 NOCL = (HC - LC) / TCC
4081 GOSUB 7800
4082 VIEW SCREEN (5, 30) - (258 * SR%, 150 * SR%), , 12
4083 WINDOW (XCEN - SX, YCEN - SY) - (XCEN + SX, YCEN + SY)
4084 VEX = 50 / (ZMAX - ZMIN)
4085 YSMIN = INT((ZMIN * VEX) / 10) * 10 + 5
4086 X0 = -30 * SR%: Y0 = (70 - YSMIN) * SR%
4088 GSC = 1
4090 S = 2: B = 2.5 * SR%
4095 FOR GCOL = 1 TO NCOL - 1
4100 FOR GROW = 1 TO NROW - 1
4110 X1 = X0 + (GCOL - GROW) * B
4115 GRC1 = (GROW + GCOL) * S
4120 Y1 = Y0 + (Z(GROW, GCOL) * VEX - GRC1) * SR%
4130 X2 = X1 - B
4135 GRC2 = (GROW + GCOL - 1) * S
4140 Y2 = Y0 + (Z(GROW, GCOL - 1) * VEX - GRC2) * SR%
4150 X3 = X1 + B
4155 GRC3 = (GROW - 1 + GCOL) * S
4160 Y3 = Y0 + (Z(GROW - 1, GCOL) * VEX - GRC3) * SR%
4170 X4 = X1
4175 GRC4 = (GROW - 1 + GCOL - 1) * S
4180 Y4 = Y0 + (Z(GROW - 1, GCOL - 1) * VEX - GRC4) * SR%
4181 ZAV = Z(GROW, GCOL) + Z(GROW, GCOL - 1) + Z(GROW - 1, GCOL)
4182 ZAV = (ZAV + Z(GROW - 1, GCOL - 1)) / 4
4183 CLOR = INT((HC - ZAV) / TCC) + 1
4190 LINE (X1, Y1) - (X2, Y2), CL%(CLOR)
4200 LINE - (X4, Y4), CL%(CLOR): LINE - (X3, Y3), CL%(CLOR)
4210 LINE - (X1, Y1), CL%(CLOR)
4220 Y = (Y1 + Y4) / 2
4230 PAINT (X4, Y), CL%(CLOR), CL%(CLOR)
```

```
4240 LINE (X1, Y1) - (X2, Y2), 15
4250 LINE - (X4, Y4), 15: LINE - (X3, Y3), 15
4260 LINE - (X1, Y1), 15
4270 NEXT GROW, GCOL
4271 NROW = NROW - 1: NCOL = NCOL - 1
4275 X1 = X0 + (-NROW) * B
4276 Y1 = Y0 + ((-NROW) * S + Z(NROW, 0) * VEX) * SRY%
4277 X2 = X0 + (-NROW) * B
4278 Y2 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4280 LINE (X1, Y1) - (X2, Y2), 15
4285 X1 = X0 + (NCOL - NROW) * B
4286 Y1 = Y0 + (-(NCOL + NROW) * S + Z(NROW, NCOL) * VEX) * SRY%
4287 X2 = X0 + (NCOL - NROW) * B
4288 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4290 LINE (X1, Y1) - (X2, Y2), 15
4295 X1 = X0 + (NCOL) * B
4296 Y1 = Y0 + ((-NCOL) * S + Z(0, NCOL) * VEX) * SRY%
4297 X2 = X0 + (NCOL) * B
4298 Y2 = Y0 + ((-NCOL) * S + YSMIN) * SRY%
4300 LINE (X1, Y1) - (X2, Y2), 15
4305 X1 = X0 + (-NROW) * B
4306 Y1 = Y0 + ((-NROW) * S + YSMIN) * SRY%
4307 X2 = X0 + (NCOL - NROW) * B
4308 Y2 = Y0 + (-(NCOL + NROW) * S + YSMIN) * SRY%
4310 LINE (X1, Y1) - (X2, Y2), 15
4317 X2 = X0 + (NCOL) * B
4318 Y2 = Y0 + (-NCOL) * S + YSMIN) * SRY%
4320 LINE - (X2, Y2), 15
4330 GOTO 9500
```

Appendix F
Menu programs

APPENDIX F.1:

SCREEN GRAPHICS MENU COMPUTER PROGRAM "PC-MENU"

This menu program was discussed in Section 6.1, and is automatically loaded after running the installation program **"PGINSTAL"** described in Section 1.7 (Appendix A.1). The user is presented with a set of menu screens to choose the methods of data entry or presentation using the function keys F1 to F10.

Screen graphics menu

(a) Initialization and control Line numbers 10–90

This first segment of code contains the copyright notice and the string variable **PCF$** required to recall the installation set up.

(b) Global menu Line numbers 100–570

This is the first menu screen and allows the user to select the type of data entry or type of chart using the function keys. This initial menu page leads to other pages to select specific types of presentation.

(c) Data entry menu Line numbers 1000–1350

This menu allows the user to enter or recall data from the three programs described in Sections 1.10.1–3, that is, **"KEYBDATA", "DISCDATA"** and **"KEYBEQUD",** as listed in Appendix A.

(d) Line/area graph menu Line numbers 2000–2400

This menu allows the user to select any of the presentation graphics formats described in Chapter 2 and listed in Appendix B.

(e) Pie/contours chart menu Line numbers 3000–3380

This menu allows the user to select any of the presentation graphics formats described in Chapter 4 and listed in Appendix D.

(f) 3D Charts/profiles menu Line numbers 4000–4380

This menu allows the user to select any of the presentation graphics formats described in Chapter 5 and listed in Appendix E.
This program also requires the following routines.

(g) Appendix A.2 Line numbers 8000–8180
(h) Appendix A.15 Line numbers 9000–9050

"PG-MENU"—BASIC program

```
10 REM <PG-MENU> Presentation Graphics MENU
14 REM (C) Copyright P.H.Milne 1991
16 REM ALL RIGHTS RESERVED
20 REM VERSION PC-1.00, 28-04-91 : Appendix Nos.
30 CLEAR
50 PCF$ = "PGSCRDSK.PGD"
60 GOSUB 8000: REM Check Screen & Disc
90 WINDOW (0, 0) - (639, 199)
100 CLS
110 IF SR% = 2 AND SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
120 LINE (10, 20) – (620, 185), RGB%, B
130 LINE (6, 18) – (624, 187), RGB%, B
140 LINE (10, 150) – (620, 150), RGB%
150 LINE (14, 152) – (616, 183), 3, BF
160 GOSUB 9000
170 FOR I = 1 TO 5
180 LINE (65, 141 – I * 16) - (100, 154 – I * 16), 12, BF
190 LINE (305, 141 – I * 16) - (340, 154 – I * 16), 12, BF
200 NEXT I
210 LOCATE 9, 10: PRINT "F1":
215 LOCATE 9, 15: PRINT "Enter DATA (Keyboard)"
220 LOCATE 9, 40: PRINT "F2":
225 LOCATE 9, 45: PRINT "Load DATA (from Disc)"
230 LOCATE 11, 10: PRINT "F3":
235 LOCATE 11, 15: PRINT "View Line/Area Graph"
240 LOCATE 11, 40: PRINT "F4":
245 LOCATE 11, 45: PRINT "View Bar Chart"
250 LOCATE 13, 10: PRINT "F5":
255 LOCATE 13, 15: PRINT "View Pie/Contours"
260 LOCATE 13, 40: PRINT "F6":
265 LOCATE 13, 45: PRINT "View 3D Charts"
270 LOCATE 15, 10: PRINT "F7":
275 LOCATE 15, 15: PRINT "View 3D Profiles"
280 LOCATE 15, 40: PRINT "F8":
285 LOCATE 15, 45: PRINT "Plot Graph/Chart"
290 LOCATE 17, 10: PRINT "F9":
295 LOCATE 17, 15: PRINT "Change PG Setup"
```

```
300 LOCATE 18, 15: PRINT "Drive - "; DDSK$
310 LOCATE 17, 40: PRINT "F10":
315 LOCATE 17, 45: PRINT "Return to MS-DOS"
320 LOCATE 20, 25: PRINT "Select Key Code - F10 to Quit";
330 LOCATE 22, 25: PRINT "(C) Copyright P.H.Milne 1991"
340 A$ = INKEY$
350 IF LEN(A$) = 2 THEN 360 ELSE 340
360 F = ASC(LEFT$(A$, 1)): S = ASC(RIGHT$(A$, 1))
370 IF S < 59 THEN 340
380 IF S > 68 THEN 340
400 ON S - 58 GOTO 410, 420, 430, 440, 450, 460, 470, 480, 490, 500
410 GOTO 1000
420 GOTO 1000
430 GOTO 2000
440 RUN MDSK$ + "GRAPH-C1"
450 GOTO 3000
460 GOTO 4000
470 GOTO 4000
480 RUN MDSK$ + "PLT-MENU"
490 RUN MDSK$ + "PGINSTAL"
500 LOCATE 24, 26: PRINT "End Graphics Session (Y/N) ?";
510 A$ = INKEY$
520 IF A$ = "" THEN 510
530 IF INSTR("YNyn", A$) = 0 THEN 510
540 IF A$ = "Y" OR A$ = "y" THEN 570
550 LOCATE 24, 25: PRINT " ";
560 GOTO 330
570 SYSTEM
1000 CLS
1010 IF SR% = 2 AND SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
1020 LINE (10, 20) - (620, 185), RGB%, B
1030 LINE (6, 18) - (624, 187), RGB%, B
1040 LINE (10, 150) - (620, 150), RGB%
1050 LINE (14, 152) - (616, 183), 3, BF
1060 GOSUB 9000
1070 FOR I = 1 TO 3
1080 LINE (65, 141 - I * 16) - (100, 154 - I * 16), 14, BF
1090 LINE (305, 141 - I * 16) - (340, 154 - I * 16), 14, BF
1100 NEXT I
1110 LOCATE 9, 10: PRINT "F1":
1115 LOCATE 9, 15: PRINT "Enter DATA (Keyboard)"
1120 LOCATE 9, 40: PRINT "F2":
1125 LOCATE 9, 45: PRINT "Load DATA (from Disc)"
1130 LOCATE 11, 10: PRINT "F3":
1135 LOCATE 11, 15: PRINT "Generate Surface DATA"
1140 LOCATE 11, 40: PRINT "F4":
1145 LOCATE 11, 45: PRINT " "
1190 LOCATE 13, 10: PRINT "F5":
1195 LOCATE 13, 15: PRINT "Change PG Setup"
```

```
1200 LOCATE 14, 15: PRINT "Drive - "; DDSK$
1210 LOCATE 13, 40: PRINT "F6":
1215 LOCATE 13, 45: PRINT "Return to MENU"
1220 LOCATE 20, 25: PRINT "Select Key Code - F6 to Return";
1230 LOCATE 22, 25: PRINT "(C) Copyright P.H.Milne 1991"
1240 A$ = INKEY$
1250 IF LEN(A$) = 2 THEN 1260 ELSE 1240
1260 F = ASC(LEFT$(A$, 1)): S
1270 IF S < 59 THEN 1240
1280 IF S > 64 THEN 1240
1300 ON S - 58 GOTO 1310, 1320, 1330
1310 RUN MDSK$ + "KEYBDATA"
1320 RUN MDSK$ + "DISCDATA"
1330 RUN MDSK$ + "KEYBEQUD"
1340 RUN MDSK$ + "PGINSTAL"
1350 GOTO 100
2000 CLS
2010 IF SR% = 2 AND SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
2020 LINE (10, 20) - (620, 185), RGB%, B
2030 LINE (6, 18) - (624, 187), RGB%, B
2040 LINE (10, 150) - (620, 150), RGB%
2050 LINE (14, 152) - (616, 183), 3, BF
2060 GOSUB 9000
2070 FOR I = 1 TO 5
2080 LINE (65, 141 - I * 16) - (100, 154 - I * 16), 14, BF
2090 LINE (305, 141 - I * 16) - (340, 154 - I * 16), 14, BF
2100 NEXT I
2110 LOCATE 9, 10: PRINT "F1":
2115 LOCATE 9, 15: PRINT "Line Graph"
2120 LOCATE 9, 40: PRINT "F2":
2125 LOCATE 9, 45: PRINT "Linear Regression"
2130 LOCATE 11, 10: PRINT "F3":
2135 LOCATE 11, 15: PRINT "Exponential Fit"
2140 LOCATE 11, 40: PRINT "F4":
2145 LOCATE 11, 45: PRINT "Power Curve Fit"
2150 LOCATE 13, 10: PRINT "F5":
2155 LOCATE 13, 15: PRINT "Cubic Spline"
2160 LOCATE 13, 40: PRINT "F6":
2165 LOCATE 13, 45: PRINT "Area Chart"
2170 LOCATE 15, 10: PRINT "F7":
2175 LOCATE 15, 15: PRINT "Stacked Area Chart"
2180 LOCATE 15, 40: PRINT "F8":
2185 LOCATE 15, 45: PRINT "100% Stacked Area"
2190 LOCATE 17, 10: PRINT "F9":
2195 LOCATE 17, 15: PRINT "Change PG Setup"
2200 LOCATE 18, 15: PRINT "Drive - "; DDSK$
2210 LOCATE 17, 40: PRINT "F10":
2215 LOCATE 17, 45: PRINT "Return to MENU"
2220 LOCATE 20, 25: PRINT "Select Key Code - F10 to Return"
```

```
2230 LOCATE 22, 25: PRINT "(C) Copyright P.H.Milne 1991"
2240 A$ = INKEY$
2250 IF LEN(A$) = 2 THEN 2260 ELSE 2240
2260 F = ASC(LEFT$(A$, 1)): S = ASC(RIGHT$(A$, 1))
2270 IF S < 59 THEN 2240
2280 IF S > 68 THEN 2240
2300 ON S - 58 GOTO 2310, 2320, 2330, 2340, 2350, 2360, 2370, 2380,
   2390, 2400
2310 RUN MDSK$ + "GRAPH-B1"
2320 RUN MDSK$ + "GRAPH-B2"
2330 RUN MDSK$ + "GRAPH-B3"
2340 RUN MDSK$ + "GRAPH-B3"
2350 RUN MDSK$ + "GRAPH-B4"
2360 RUN MDSK$ + "GRAPH-B5"
2370 RUN MDSK$ + "GRAPH-B5"
2380 RUN MDSK$ + "GRAPH-B5"
2390 RUN MDSK$ + "PGINSTAL"
2400 GOTO 100
3000 CLS
3010 IF SR% = 2 AND SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
3020 LINE (10, 20) - (620, 185), RGB%, B
3030 LINE (6, 18) - (624, 187), RGB%, B
3040 LINE (10, 150) - (620, 150), RGB%
3050 LINE (14, 152) - (616, 183), 3, BF
3060 GOSUB 9000
3070 FOR 1 = 1 TO 3
3080 LINE (65, 141 - I * 16) - (100, 154 - I * 16), 14, BF
3090 LINE (305, 141 - I * 16) - (340, 154 - I * 16), 14, BF
3100 NEXT I
3110 LOCATE 9, 10: PRINT "F1":
3115 LOCATE 9, 15: PRINT "Pie Chart"
3120 LOCATE 9, 40: PRINT "F2":
3125 LOCATE 9, 45: PRINT "Dual Pie Chart"
3130 LOCATE 11, 10: PRINT "F3":
3135 LOCATE 11, 15: PRINT "Line Contours"
3140 LOCATE 11, 40: PRINT "F4":
3145 LOCATE 11, 45: PRINT "Shaded Contours"
3170 LOCATE 13, 10: PRINT "F5":
3175 LOCATE 13, 15: PRINT "Change PG Setup"
3180 LOCATE 14, 15: PRINT "Drive - "; DDSK$
3190 LOCATE 13, 40: PRINT "F6":
3195 LOCATE 13, 45: PRINT "Return to MENU"
3220 LOCATE 20, 25: PRINT "Select Key Code - F6 to Return";
3230 LOCATE 22, 25: PRINT "(C) Copyright P.H.Milne 1991"
3240 A$ = INKEY$
3250 IF LEN(A$) = 2 THEN 3260 ELSE 3240
3260 F = ASC(LEFT$(A$, 1)): S = ASC(RIGHT$(A$, 1))
3270 IF S < 59 THEN 3240
3280 IF S > 66 THEN 3240
```

```
3300 ON S - 58 GOTO 3310, 3320, 3330, 3340, 3370, 3380
3310 RUN MDSK$ + "GRAPH-D1"
3320 RUN MDSK$ + "GRAPH-D1"
3330 RUN MDSK$ + "GRAPH-D2"
3340 RUN MDSK$ + "GRAPH-D3"
3370 RUN MDSK$ + "PGINSTAL"
3380 GOTO 100
4000 CLS
4010 IF SR% = 2 AND SCR% = 2 THEN RGB% = 12 ELSE RGB% = 3
4020 LINE (10, 20) - (620, 185), RGB%, B
4030 LINE (6, 18) - (624, 187), RGB%, B
4040 LINE (10, 150) - (620, 150), RGB%
4050 LINE (14, 152) - (616, 183), 3, BF
4060 GOSUB 9000
4070 FOR I = 1 TO 4
4080 LINE (65, 141 - I * 16) - (100, 154 - I * 16), 14, BF
4090 LINE (305, 141 - I * 16) - (340, 154 - I * 16), 14, BF
4100 NEXT I
4110 LOCATE 9, 10: PRINT "F1":
4115 LOCATE 9, 15: PRINT "3D Graph"
4120 LOCATE 9, 40: PRINT "F2":
4125 LOCATE 9, 45: PRINT "3D Area Chart"
4130 LOCATE 11, 10: PRINT "F3":
4135 LOCATE 11, 15: PRINT "Line Profile"
4140 LOCATE 11, 40: PRINT "F4":
4145 LOCATE 11, 45: PRINT "3D Open Mesh"
4150 LOCATE 13, 10: PRINT "F5":
4155 LOCATE 13, 15: PRINT "3D Line Profile"
4160 LOCATE 13, 40: PRINT "F6":
4165 LOCATE 13, 45: PRINT "3D Shaded Mesh"
4170 LOCATE 15, 10: PRINT "F7":
4175 LOCATE 15, 15 PRINT "Change PG Setup"
4180 LOCATE 16, 15: PRINT "Drive -"; DDSK$
4190 LOCATE 15, 40: PRINT "F8":
4195 LOCATE 15, 45: PRINT "Return to MENU"
4220 LOCATE 20, 25: PRINT "Select Key Code - F8 to Return";
4230 LOCATE 22, 25: PRINT "(C) Copyright P.H.Milne 1991"
4240 A$ = INKEY$
4250 IF LEN(A$) = 2 THEN 4260 ELSE 4240
4260 F = ASC(LEFT$(A$, 1)): S = ASC(RIGHT$(A$, 1))
4270 IF S < 59 THEN 4240
4280 IF S > 66 THEN 4240
4300 ON S - 58 GOTO 4310, 4320, 4330, 4340, 4350, 4360, 4370, 4380
4310 RUN MDSK$ + "GRAPH-E1"
4320 RUN MDSK$ + "GRAPH-E1"
4330 RUN MDSK$ + "GRAPH-E2"
4340 RUN MDSK$ + "GRAPH-E4"
4350 RUN MDSK$ + "GRAPH-E3"
4360 RUN MDSK$ + "GRAPH-E5"
```

```
4370 RUN MDSK$ + "PGINSTAL"
4380 GOTO 100
```

APPENDIX F.2:

PLOTTER MENU COMPUTER PROGRAM "PLT-MENU"

This program is discussed in Section 6.1 and is called from the screen graphics menu described in Appendix F.1. The layout of the program is identical to **"PG-MENU",** where the plotter programs to be called require a "GPLOT" prefix rather than a "GRAPH" prefix. Note that not all the screen graphics programs can be transferred to a plotter, as listed in Table 6.1.

The following changes are required to Appendix F.1 to create the program **"PLT-MENU":**

replace "PG-MENU" with "PLT-MENU" in Line 10
replace "Plot Graph/Chart" with "Return to VIEW Menu" in Line 280
replace "GRAPH-C1" with "GPLOT-C1" in Line 440
replace "PLT-MENU" with "PG-MENU" in Line 480
replace "GRAPH-B1" with "GPLOT-B1" in Line 2310
replace "GRAPH-B2" with "GPLOT-B2" in Line 2320
replace "GRAPH-B3" with "GPLOT-B2" in Line 2330
replace "GRAPH-B3" with "GPLOT-B2" in Line 2340
replace "GRAPH-B4" with "GPLOT-B4" in Line 2350
replace "GRAPH-B5" with "GPLOT-B5" in Line 2360
replace "GRAPH-B5" with "GPLOT-B5" in Line 2370
replace "GRAPH-B5" with "GPLOT-B5" in Line 2380
replace "GRAPH-D1" with "GPLOT-D1" in Line 3310
replace "GRAPH-D1" with "GPLOT-D1" in Line 3320
replace "GRAPH-D2" with "GPLOT-D2" in Line 3330
replace "GRAPH-E4" with "GPLOT-E4" in Line 4340
replace "GRAPH-E3" with "GPLOT-E3" in Line 4350
replace "PG-MENU" with "PLOT-MENU" in Line 9000.

References

Batty, M. (1987) *Microcomputer Graphics,* Chapman and Hall, London.

Boyle, C. (1986) *Mastering Statistics with your microcomputer,* Macmillan, Basingstoke.

British Standards Institution (1977) *Recommendation for the presentation of tables, graphs and charts,* (DD 52:1977), London.

Ford, N. (1984) *Business Graphics for the IBM PC,* SYBEX Inc., Berkeley.

Hannagan, T.J. (1986) *Work Out Statistics 'O' Level and GCSE,* Macmillan Master Series, London.

Harris, D. (1984) *Computer Graphics and Applications,* Chapman and Hall, London.

Institute of Hydrology (1985) *Hydrological Data UK, 1982 Yearbook,* Wallingford.

Laflin, S. (1987) *Two-dimensional Computer Graphics,* Chartwell-Bratt, Bromley.

McKenzie, J. (ed) (1978) *Interactive Computer Graphics in Science Teaching,* Ellis Horwood Limited, Chichester.

Milne, P.H. (1987) *Computer Graphics for Surveying,* E. & F.N. Spon Ltd, London.

National Audit Office (1989) *Quality control of road and bridge construction,* H.M.S.O., London.

Orvis, W.J. (1987) *1-2-3 for Scientists & Engineers,* SYBEX Inc., Alameda.

Reynolds, L. and Simmonds, D. (1984) *Presentation of Data in Science,* Martinus Nijhoff Publishers, Dordrecht.

Rowell, J. (1988) *Picture Perfect: Colour Output in Computer Graphics,* Tektronix, Beaverton.

Sandler, C. (1986) *Business Graphics on the IBM PC,* Addison Wesley Publishing Company, Wokingham.

Savic, D. (1985) *BASIC Interactive Graphics,* Butterworths, London.

Simmonds, D. (ed.) (1980), *Charts & Graphs,* MTP Press Limited, Lancaster, Baltimore.

Spear, M.E. (1978) *Practical Charting Techniques,* McGraw-Hill Book Company, New York.

Turner, B.T. (1990) Effective Technical Writing and Speaking, *The Manual,* ADMARK (Publishing) Ltd., Vol 3(3), pp. 4–8.

Waite, M. and Morgan, C.L. (1983) *Graphics primer for the IBM PC,* Osborne/McGraw-Hill, Berkeley.

Which? (1989) Car Safety—the facts you need to know, Consumers' Association Ltd., October, pp 489–93.

Wolfram, S. (1988) *Mathematica,* Addison-Wesley Publishing Company, Redwood City.

Index