Mathematical Methods in Computer Graphics and Design

K. W. BRODLIE

Mathematical Methods in Computer Graphics and Design

Based on the proceedings of the conference on Mathematical Methods in Computer Graphics and Design, organized by the Institute of Mathematics and its Applications and held at the University of Leicester on September 28th, 1978

Edited by

K. W. BRODLIE

Computer Laboratory
University of Leicester
Leicester, England

1980

ACADEMIC PRESS

A Subsidiary of Harcourt Brace Jovanovich, Publishers

London New York Toronto Sydney San Francisco

ACADEMIC PRESS INC. (LONDON) LTD.
24/28 Oval Road,
London NW1

United States Edition published by
ACADEMIC PRESS INC.
111 Fifth Avenue
New York, New York 10003

British Library Cataloguing in Publication Data
Mathematical Methods in Computer Graphics and Design (*Conference*), *University of Leicester, 1978*
Mathematical methods in computer graphics and design.
1. Electronic data processing—Congresses
I. Brodlie, K. W. II. Institute of Mathematics and its Applications
519.4 QA75.5 79-50302

ISBN 0-12-134880-6

Printed in Great Britain by
Whitstable Litho Ltd., Whitstable, Kent

CONTRIBUTORS

I. C. BRAID; *Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge, CB2 3QG.*

K. W. BRODLIE; *Computer Laboratory, University of Leicester, Leicester, LE1 7RH.*

A. R. FORREST; *School of Computing Studies, University of East Anglia, University Plain, Norwich, NOR 88C.*

R. C. HILLYARD; *Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge, CB2 3QG.*

D. H. McLAIN; *Computer Services, University of Sheffield, The Hicks Building, Sheffield, S10 2TN.*

M. A. SABIN; *Kongsberg Ltd., Data Systems Division, St. Peter's Road, Maidenhead, Berks., SL6 7QU. Now at CAD Centre, Cambridge.*

I. A. STROUD; *Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge, CB2 3QG.*

D. C. SUTCLIFFE; *Interactive Computing and Graphics Group, Atlas Computing Division, Rutherford Laboratory, Chilton, Oxfordshire, OX11 0QX.*

PREFACE

This book contains the proceedings of a conference "Mathematical Methods in Computer Graphics and Design", organized by the Institute of Mathematics and its Applications and held at the University of Leicester on 28th September, 1978.

The idea for the conference came from the IMA Numerical Analysis group, who felt there was a need to bring together those developing graphical algorithms and those likely to use them. The size and range of the audience confirmed the wide interest in computer graphics - over 200 people attended, with a good balance between representatives from universities, government research establishments and private industry.

The speakers, too, represented a wide range of interests and included specialists in numerical analysis, computer graphics and computer-aided design. The order of the papers in this volume reflects the order in which the talks were presented.

The first paper, by myself, gives a review of methods for curve and function drawing. Two papers on contouring follow, one by Dale Sutcliffe (Rutherford Laboratory) describing the regular grid case and the other by Malcolm Sabin (CAD Centre) reviewing methods for scattered data. Audience reaction at the conference showed that many people from different backgrounds are interested in contouring, and that continuing efforts to develop good contouring algorithms are certainly justified. The fourth paper is by Dermot McLain (University of Sheffield), who discusses the problems of curve and surface drawing when the data are subject to errors, and points out the need to embed computer graphics techniques in more complex systems where information from databases is used to supplement the data supplied by the user. The final two papers by Robin Forrest (University of East Anglia) and Ian Braid (University of Cambridge) turn the spotlight on computer-aided design. Forrest describes recent progress in geometric algorithms, while Braid discusses some of the problems in volume modelling.

There was a good deal of interesting discussion after each talk. Although it proved impractical to record the discussions, the audience were asked to send in their questions and comments in writing after the conference. There was a good response to this request. For example, I received several interesting questions on my talk and indeed one written comment brought to light a curve drawing method which performs better than any I talked about at the conference! In most cases, the questions and comments (with replies from the speakers) have been included at the end of each paper, although in some cases the authors have incorporated the comments directly into their text.

I hope the book will be of interest to those developing computer graphics algorithms, and more importantly, to those who need to apply these algorithms in practical situations. The conference was held at a time when NAG are planning the development of a chapter of graphical routines to add to their main numerical library. It is likely that early contributions to the chapter will be in the areas of curve and function drawir and contouring. The interest shown by the audience in these two areas confirmed the need for this new NAG development, and gave those planning the graphics chapter some indication of particular user requirements.

There are many people to thank for their help in the organisation of the conference. In particular it is a pleasure to give special thanks to three people: Catherine Richards of the IMA, for her overall help, Cacs Hinds of the IMA, for her hard work "behind-the-scenes" in organizing the conference and Geoff Hayes for acting as Chairman and making sure the program ran smoothly.

Finally I wish to thank all those who have helped in the preparation of these proceedings. Particular thanks are due to my colleague Andrew Nash for his help with artwork, and to my wife, Trish, for her editorial assistance. Finally I would like to thank Miss J. Fulkes and Mrs. S. Hockett of the IMA for their accurate typing of the final manuscript.

November 1979

Kenneth W. Brodlie

ACKNOWLEDGEMENTS

The Institute thanks the authors of the papers, the editor, Dr. K. W. Brodlie (University of Leicester) and also Miss Janet Fulkes and Mrs. Susan Hockett for typing the papers.

CONTENTS

1. A REVIEW OF METHODS FOR CURVE AND FUNCTION DRAWING

K.W. Brodlie
(University of Leicester)

1. INTRODUCTION

This paper reviews methods for curve and function drawing in computer graphics. The approach taken is influenced by the author's own particular experience which lies in a general university computing service environment rather than in a specialist computer-aided design department. Thus the paper concentrates on methods for drawing a smooth curve through a number of data points - these data points typically being the result of some scientific experiment. Only a brief introduction is given to the interactive design of curves, and interested readers are directed towards the CAD literature.

The major part of the paper, then, considers the construction of a smooth curve through a number of data points (x_i, y_i), $i = 1,2,..n$. Two quite distinct cases are identified. First when it is known that there is some function underlying the data which is <u>single-valued</u> - for example, when the data represents measurements of temperature at regular time intervals. It is obviously essential that the drawn curve should also be single-valued. The usual approach is to construct some function $y = f(x)$ which interpolates the data - generally a piecewise cubic polynomial - and then plot the function $f(x)$.

There are cases of course when single-valued curves are too restrictive. Certainly when one is concerned with drawing shapes rather than plotting graphs, the possibility of a curve being multi-valued is essential. Here a different approach is needed, one in which x and y are considered separately as functions of a parameter t. The data points (x_i, y_i) are assigned parameter values t_i according to some scheme, and interpolants $x(t), y(t)$ are constructed such that $x(t_i) = x_i$, $y(t_i) = y_i$, $i = 1,2,..n$. The resulting curve $(x(t), y(t))$ is known as a <u>parametric</u> curve, and can be multi-valued, even closed. When drawing shapes, it is essential that the curves should be independent of the particular axis system used to define the data points. Thus $x(t), y(t)$ must be defined in a suitable symmetric manner to ensure that the drawn curve is independent of an axis rotation.

Both types of curve drawing are needed. Single-valued curves cannot offer the flexibility required in some situations, but equally, parametric curves which are invariant under rotation cannot guarantee to produce a single-valued curve from "single-valued data", i.e. data with x-values satisfying $x_1 < x_2 < .. < x_n$. It is surprising that GHOST (GHOST User Manual, 1978) and GINO-F (GINO-F User Manual, 1975), probably the two most widely used graphics packages in the UK, only offer one type of method - in each case a rotation-invariant parametric curve drawing routine. Of course this is disastrous for users who wish to ensure that a curve is single-valued. Figs. 1 and 2 show the curves drawn by the two packages through a set of data points, representing measurements of the speed of a particle at regular time intervals. No further comment is really needed - the resulting curves simply do not make sense.

Various aspects of single-valued curve drawing are described in sections 2 - 4 of this paper, and parametric curves are discussed in section 5.

Throughout this paper it is assumed that the curve is to be drawn through the data points. Often, however, the data points are recognised to be subject to error, and the user simply wishes the curve to approximate the data, say in a least-squares sense. This case is not discussed in this paper, since it is dealt with in McLain's paper, Chapter 4 of this book.

Two other topics, however, are discussed. Section 6 gives a brief introduction to the interactive design of curves, and relates the methods used by designers to create curves of a desired shape, to the methods for drawing curves through data points described in the earlier sections of the paper. Finally, in section 7, the problem of plotting a user-supplied function of one variable is discussed.

2. SINGLE-VALUED CURVES - CUBIC SPLINES

Probably the best known technique for constructing a single-valued curve $y = f(x)$ through the data points (x_i, y_i), $i = 1,2,..n$, is that of spline interpolation. A spline is simply a piecewise polynomial of degree m with its first (m- 1)

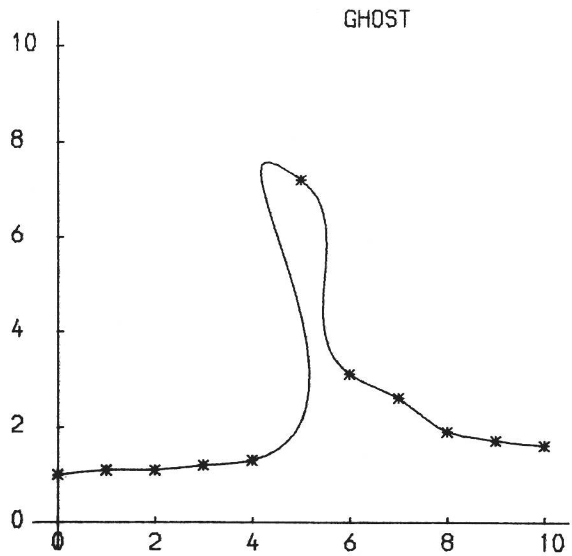

Fig. 1

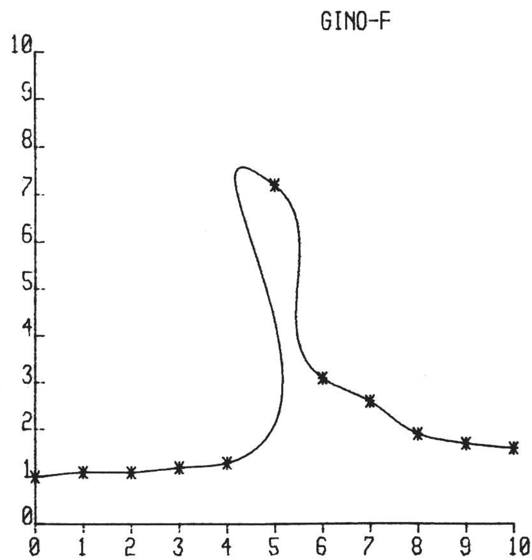

Fig. 2

derivatives continuous at the joins. More precisely, a spline function $s(x)$ of degree m with knots $\lambda_1, \lambda_2, .. \lambda_k$ $(\lambda_1 < \lambda_2 .. < \lambda_k)$ is a function with the following properties:

(i) in each interval

$x \leqslant \lambda_1;\ \lambda_i \leqslant x \leqslant \lambda_{i+1}, \quad i = 1,..k - 1;\ x \geqslant \lambda_k$

$s(x)$ is a polynomial of degree m at most;

(ii) $s(x)$ and its first $(m - 1)$ derivatives are continuous.

In practice, cubic splines $(m = 3)$ are most commonly used, the second derivative continuity they provide being adequate for most situations.

For computational purposes, a cubic spline is best represented as a linear combination of B - splines, sometimes called fundamental splines. A cubic B - spline is itself a cubic spline, with the same set of knots as the original

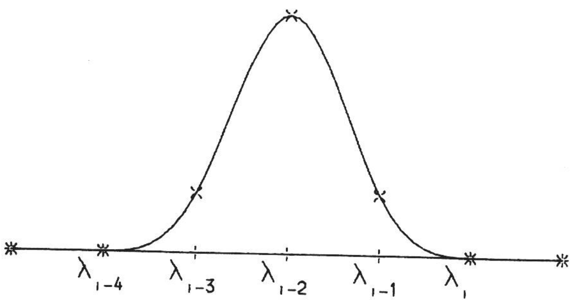

Fig. 3

spline, but with the special characteristic that it is zero everywhere except over four adjacent knot intervals. Thus the cubic B - spline $N_i(x)$ is defined as a cubic spline which is non-zero over the interval $\lambda_{i-4} < x < \lambda_i$, and zero elsewhere (see Fig. 3). In fact this is sufficient to define $N_i(x)$ uniquely, apart from an arbitrary scale factor which can be chosen so that:

$$\sum_i N_i(x) = 1$$

at all points x in the interval $\{a,b\}$ over which the spline is defined. The functions $N_i(x)$ are often called <u>normalized B - splines</u>.

A cubic spline with k knots can be expressed as a linear combination of (k+4) cubic B-splines:

$$s(x) = \sum_{i=1}^{k+4} \alpha_i N_i(x)$$

To complete the definitions of the B-splines, four additional knots are added at each end of the spline - $\lambda_{-3}, \lambda_{-2}, \lambda_{-1}, \lambda_0$ at the left-hand end and $\lambda_{k+1}, \lambda_{k+2}, \lambda_{k+3}, \lambda_{k+4}$ at the right-hand end.

The particular problem of cubic spline interpolation is to find s(x) such that:

$$s(x_j) = y_j, \; j=1,2,..n$$

$$\text{i.e.} \quad \sum_{i=1}^{k+4} \alpha_i N_i(x_j) = y_j, \; j=1,2,..n \qquad (2.1)$$

where s(x) has knots $\lambda_1, \lambda_2, .. \lambda_k$. Notice that the system (2.1) has n equations in (k + 4) unknowns α_i. An obvious strategy is to select the central (n - 4) interior data points as the knots of the spline, giving a fully determined system of n equations in n unknowns.

An alternative is to choose all the (n - 2) interior data points as knots, and this gives the freedom to specify two conditions in addition to the n interpolatory conditions (2.1). In particular, it allows the slopes of the spline at the end-points x_1 and x_n to be specified, should these happen to be known - this gives what is termed a clamped spline. Alternatively, the end-conditions

$$\frac{d^2s}{dx^2} = 0, \; x = x_1 \text{ and } x = x_n$$

can be specified; this leads to the so-called natural spline which has the smoothness property that among all functions f(x) which have continuous second derivatives and pass through the data points, it minimizes

$$\int_{x_1}^{x_n} \{\frac{d^2f}{dx^2}\}^2 \, dx$$

The particular spline chosen will depend on the application, but in the absence of any special circumstances, the simple strategy of selecting the central (n - 4) data points as knots usually works well. The advantage of the B - spline representation is evident in the solution of the equations (2.1): since each B - spline function $N_i(x)$ is zero nearly everywhere, the equations have a convenient banded structure. It is important from a numerical standpoint that the B - splines are evaluated accurately and efficiently - see Cox [1972].

It has only been possible here to give a brief outline of spline interpolation. Good references on cubic spline interpolation in the numerical analysis literature are the papers by Cox [1975, 1977]; a good reference describing the various end-conditions for cubic splines in CAD is the paper by Adams [1974].

Two main objections have been levelled at cubic splines in the curve drawing context. The first is their tendency to produce unwanted points of inflection in the curve. Fig. 4 shows the cubic spline interpolant for a set of cost-effectiveness data. For theoretical reasons, the curve is known to be everywhere concave and free from inflection points - so the spurious inflection point introduced by the spline interpolant is misleading. Fig. 5 shows another example of the rather 'loose' type of curve generated by cubic splines - notice the 'overshoots' at either side of the main peak.

It is helpful at this stage to think of the spline as a thin strip of wood or plastic constrained to pass through the data points. The unwanted inflection points can be removed by pulling on the ends of the strip. If sufficient

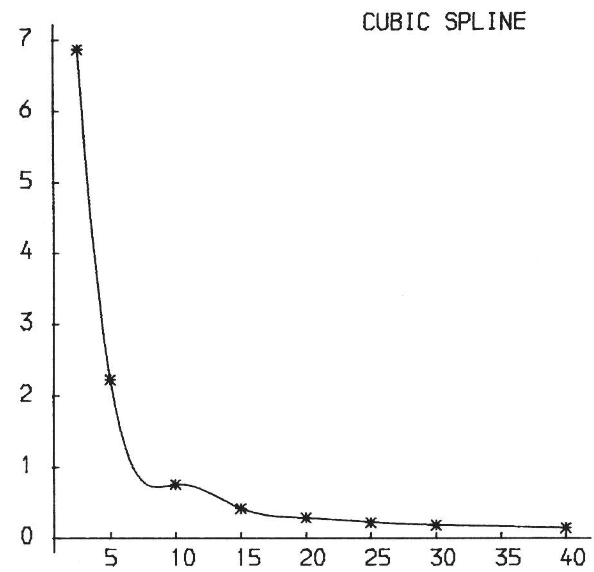
CUBIC SPLINE
7
6
5
4
3
2
1
0
5
10
15
20
25
30
35
40

Fig. 4

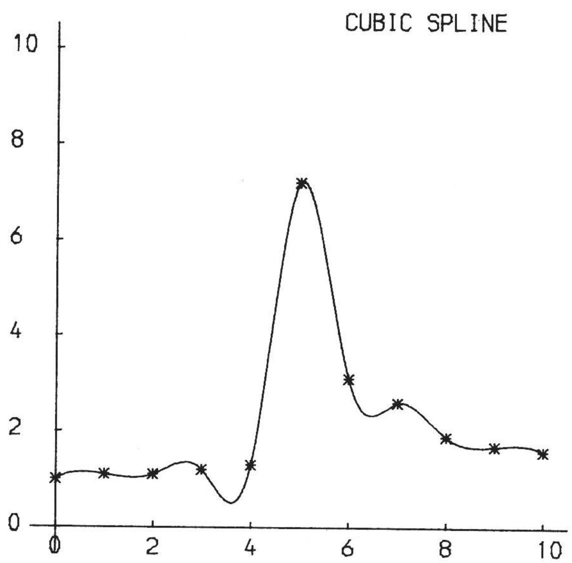
CUBIC SPLINE
10
8
6
4
2
0
0
2
4
6
8
10

Fig. 5

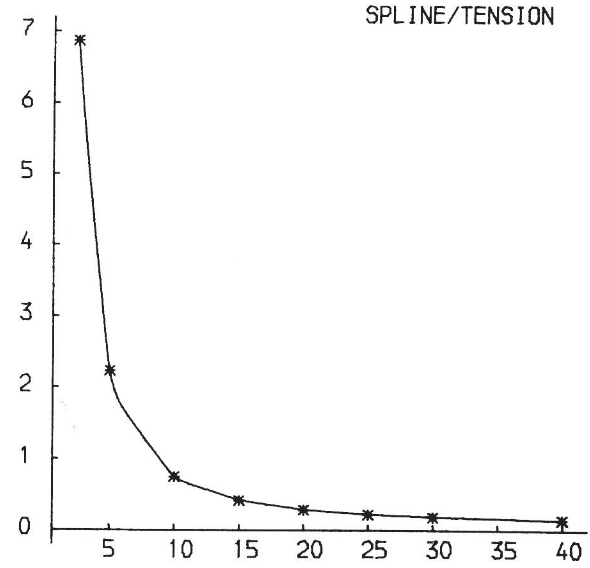
SPLINE/TENSION
7
6
5
4
3
2
1
0
5
10
15
20
25
30
35
40

Fig. 6

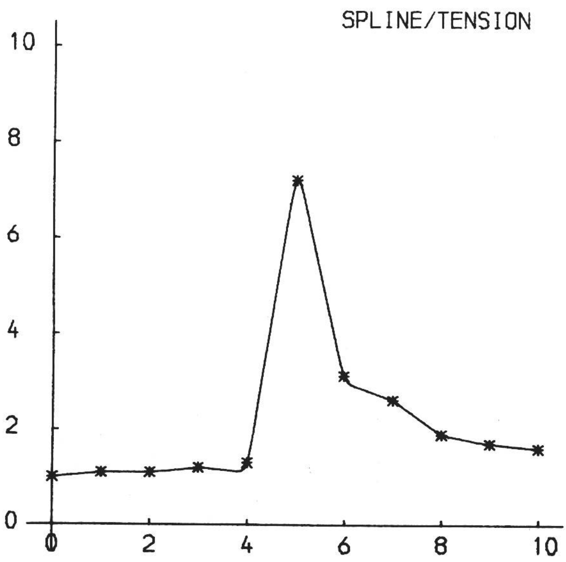
SPLINE/TENSION
10
8
6
4
2
0
0
2
4
6
8
10

Fig. 7

tension is applied, the data points will simply be connected by straight line pieces. This notion has been translated into mathematical terms in the form of a <u>spline under tension</u>, suggested originally by Schweikert [1966] and developed later by Cline [1974] and Pilcher [1974].

Cline's presentation is followed here. He seeks a function f(x) such that:

$$f(x_i) = y_i, \quad i = 1,2,..n$$

and

$$f^{(2)}(x) - \sigma^2 f(x) \quad (\text{where } f^{(2)}(x) = \frac{d^2f}{dx^2})$$

is continuous in $\{x_1, x_n\}$ and linear on each subinterval $\{x_i, x_{i+1}\}$, $i = 1,2,..n-1$. The factor σ is known as the tension factor: if $\sigma = 0$, the function f is simply a cubic spline, while as $\sigma \to \infty$, the function f tends to a piecewise linear function connecting the data points. The intention is that as σ increases from zero, so the curve defined by f should appear to give a 'tighter' fit to the data.

The condition that $(f^{(2)}(x) - \sigma^2 f(x))$ be linear on each subinterval leads to a set of ordinary differential equations. These are easily solved (see Cline's paper for details), giving the spline under tension as:

$$\begin{aligned} f(x) = {} & \frac{f^{(2)}(x_i)}{\sigma^2} \cdot \frac{\sinh(\sigma(x_{i+1}-x))}{\sinh(\sigma h_i)} \\ & + \left(y_i - \frac{f^{(2)}(x_i)}{\sigma^2}\right) \frac{(x_{i+1}-x)}{h_i} + \frac{f^{(2)}(x_{i+1})}{\sigma^2} \cdot \frac{\sinh(\sigma(x-x_i))}{\sinh(\sigma h_i)} \\ & + \left(y_{i+1} - \frac{f^{(2)}(x_{i+1})}{\sigma^2}\right) \frac{(x-x_i)}{h_i} \end{aligned} \qquad (2.2)$$

for x in the interval $\{x_i, x_{i+1}\}$. Here $h_i = x_{i+1} - x_i$. The unknown second derivatives $f^{(2)}(x_i)$ are found by differentiating (2.2) and matching $f^{(1)}(x)$ at the end-points of intervals - a tridiagonal system of equations has to be solved.

The curves produced by splines under tension for the