tanulmányok

62/1977

MTA Számitástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

GEOMETRY AND ALGEBRA SOME PAPERS

by

S.A. Coons

Tanulmányok 62/1977.

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 040 8

778716 MTA KESZ Sokszorosító. F. v.: Szabó Gyula

TARTALOMJEGYZÉK

Surface pathes 7
Bezier curves and surfaces, with applications to blending functions. 37
Remarks about B-splines 50
The modification of the shape of piece-wise curves 63
Photogrammetric measurement 73
Products of polynomials 83
Constrained least-squares 85

Jelen dolgozat az R-12-es számú intézeti témában került kidolgozásra

GEOMETRY AND ALGEBRA SOME PAPERS BY S.A. COONS

This first Budapest volume of Coonsiana is a selection from Steve's voluminous output of papers since he has been with us. They are concerned mainly with his beloved surfaces and curves, but also evidence his wide-ranging activities in other, related realms of mathematics. By the time they go to press, the material for a further volume has piled up.

Let these volumes be a token of our esteem for Steve, with whom it is now our privilege to work together.

J. Hatvany
Head of Mechanical Engineering
Automation Division

SURFACE PATCHES.

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

This will be a brief review of the idea of quite general (but not completely general) surface description by mathematical means.

The original idea was invented in about 1946, thirty years ago, but it was of not much use then because as you will see it involved a burden of calculation that in those days was insupportable. We had at that time only desk-calculators that did their arithmetic entirely mechanically, and it was simply unheard of to "program" such a machine.

[It used to be fun to direct such a machine to divide one by zero. It worked faithfully at the task until it was shut off. Nowadays, my tiny pocket calculator, no bigger than a package of cigarettes, instantly says "Error" when I ask it to do such a ridiculous calculation.]

So the idea of these surface patches was shelved until about 1964, when computers were programmable, and lightning fast, and thought nothing of gobbling up numbers and batting them about and spitting out answers faster than humans could assimilate them. In America, we called these devices "number-crunchers".

But this is not intended as a history of computers, and these remarks only show that the idea of general "free-from" mathematical surface description had to await the computer to be in any sense a "practical" idea.

I have become accustomed to a particular personal notation, peculiar to me, which at first sight may be repugnant to the reader. Hopefully, he will after a little while realize that my notation is not so terribly bad after all. I can only recall the classic remark: "Suspend dis-belief".

I exhibit a very simple surface (patch) equation:

$$uv = \begin{bmatrix} F_0u & F_1u \end{bmatrix} \begin{bmatrix} 0v \\ 1v \end{bmatrix} + \begin{bmatrix} u0 & u1 \end{bmatrix} \begin{bmatrix} F_0v \\ F_1v \end{bmatrix} - \begin{bmatrix} F_0u & F_1u \end{bmatrix} \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \begin{bmatrix} F_0v \\ F_1v \end{bmatrix}$$

Here, to explain the notational conventions, one should regard the "bi-literal" symbol uv as standing for a vector function of the two "independent" variables u and v.

The fact that these two letters are adjacent does <u>not</u> mean that they are to be multiplied together.

It is easy, but cumbersome, to write

$$\left[\; x \;\; y \;\; z \;\right] + \left[f(u,v) \;\; g(u,v) \;\; h(u,v) \right]$$

This is the accepted kind of notation that my bi-literal symbol uv means. It is orthodox, while my notation is heresy.

"A surface is the locus of a point moving in space with two degrees of freedom."

The two degrees of freedom are measured by the values u and v, sometimes called the "parameters".

[Later on, we will see that we can introduce more "degrees of freedom".]

But here I must make a strong distinction between the "space of immersion" of the locus of the moving point, and the number of independent variables, (for the moment, u and v).

We think of an ordinary, proper surface as being "immersed" in a three-dimensional space. But the two degrees of freedom, u and v, permit us to make as many or as few vector functions of these two directing variables as we please. I cannot too strongly emphasize that the number of degrees of freedom and the dimension of the space of immersion are strictly independent.

It is not un-thinkable, for example, that

$$\begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} f(u,v) \end{bmatrix}.$$ (A point dancing on a line.)

or

$$\begin{bmatrix} x & y \end{bmatrix} = \begin{bmatrix} f(u,v) & g(u,v) \end{bmatrix}.$$ (A point dancing on a blackboard.)

or (skipping a three-space,)

$$\begin{bmatrix} x & y & z & w \end{bmatrix} = \begin{bmatrix} f(u,v) & g(u,v) & h(u,v) & i(u,v) \end{bmatrix}$$

and so on.

Now if one or the other of the two variables u and v are held fixed, say for example that u is held fixed with a value of zero, the bi-literal symbol becomes 0v. This describes the vector-valued function with a <u>single</u> degree of freedom.

("A curve is the locus of a point moving in space with a single degree of freedom.")

This hopefully serves to indicate the meaning of the bi-literal symbols 0v, 1v, u0 and u1, appearing in the surface equation. They are simply curves (we call them the "boundary curves") of the patch. (A more orthodox expression would be "surface segment", instead of "patch".) Each of these vector quantities describes the locus of a point moving with a single degree of freedom, either u or v.

It is a mere matter of expediency to set the other of the two variables equal to zero or one. It simplifies the arithmetic, without loss of generality. It is the customary way to deal with "piece-wise" functions.

So in orthodox notation, we could write

$$\begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} f(0,v) & g(0,v) & h(0,v) \end{bmatrix}$$

when in our more simplified notation, we write only 0v.

[I hope the reader is beginning to understand my reasons for inventing this notation .]

The vector quantities 00 01 10 and 11 appearing in the equation are exactly what they mean. 00, for example, is the vector valued quantity

$$\begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} f(0,0) & g(0,0) & h(0,0) \end{bmatrix}$$

and it describes a point, with no degrees of freedom whatever (poor thing).

Implicit in this is the so-called "compatibility condition". It says that the curve u0 and the curve 0v, for example, have a single point in common when $u = 0$ and $v = 0$. This is merely a statement that the four boundary curves must make a "closed" region, that they indeed intersect and join at the four corners, 00 01 10 and 11.

We have yet to describe the symbols F_0u, F_1u and F_0v, F_1v. These are scalar-valued functions of their variables. In more orthodox notation, we should write $F_0(u)$ for example, but there is no harm in omitting the parantheses.

For a beginning, we shall impose some very weak conditions on these two functions. Later on, we will strengthen these conditions, but for the moment, the conditions are

$$F_0(0) = 1$$
$$F_0(1) = 0$$
$$F_1(0) = 0$$
$$F_1(1) = 1$$

All of this can be more compactly described by the Kronecker delta symbols:

$$F_i j = \delta_{ij}$$

for when the index and the argument of F are alike, F takes the value one. Otherwise, it takes the value zero.

[Later on, we shall make much better use of the Kronecker delta symbols.]

In passing, we observe that in particular the functions

$$F_0 u = 1 - u$$
$$F_1 u = u$$

satisfy these conditions. But the conditions are extremely weak, and these two linear expressions are by no means the whole story, as we shall see.

But for the moment, let us return to the surface equation. It is, as a reminder,

$$uv = \begin{bmatrix} F_0 u & F_1 u \end{bmatrix} \begin{bmatrix} 0v \\ 1v \end{bmatrix} + \begin{bmatrix} u0 & u1 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

$$- \begin{bmatrix} F_0 u & F_1 u \end{bmatrix} \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

Let us see what happens when we substitute 0 for u. We have:

$$uv\Big|_{u=0} = \begin{bmatrix} F_0 0 & F_1 0 \end{bmatrix} \begin{bmatrix} 0\ v \\ 1\ v \end{bmatrix} + \begin{bmatrix} 00 & 01 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

$$- \begin{bmatrix} F_0 0 & F_1 0 \end{bmatrix} \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

Then

$$uv\Big|_{u=0} = 0v + \begin{bmatrix} 00 & 01 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

$$- \begin{bmatrix} 00 & 01 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

and finally, $$uv\big|_{u=0} = 0v$$

This means that the uv surface contains the boundary curve 0v. It means that the uv surface is partly defined by the 0v curve, and it suggests, surely, that similarly the surface contains and in some sense is defined by its other three boundary curves as well.

Of course the nature of the F-functions also plays a large part in the description of the surface. The surface just described, with the linear functions chosen as simply as possible

$$F_0u = 1-u$$
$$F_1u = u$$

is called "linearly blended". The term "blended" is strictly my own invention. I had the intuitive notion that the boundaries should be <u>somehow</u> mixed together, "blended", so that anywhere within the four-curved region, a point on the surface would owe its position <u>somehow</u> to the boundary curves.

Even so, it is striking to realize that the exact nature of the four boundaries is nowhere implicit in the equation. This is to say that the equation represents a non-denumerable interpolation scheme for the non-denumerable boundary curves. From a mathematical point of view, this is an unusual situation indeed. It has been called "trans-finite interpolation" by GORDON and others.

We will now introduce some further stipulations on the nature of the F functions.

Written out, they are

$F_0\Theta = 1$

$F_0 1 = 0$ as before, but now we add:

$F_1 0 = 0$

$F_1 1 = 1$

$F'_0 0 = 0$ The prime-marks are intended to

$F'_0 1 = 0$ indicate differentation with

$F'_1 0 = 0$ respect to the argument variable.

$F'_1 1 = 0$ They indicate the "slopes" of the functions in question.

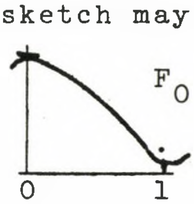
In Kronecker delta notation, this is, compactly,

$$F_i j = \delta_{ij}$$

$$F'_i j = 0.$$

Even now, the stipulations on the nature of the F functions is weak, although a little stronger than before.

Functions that can satisfy these conditions can be cubics.
A sketch may be helpful.



u, the independent variable.

The sketched function F_0 satisfies the conditions that

$F_0 0 = 1$, $F_0 1 = 0$, and also the new conditions:

$F'0 = 0$, $F'1 = 0$.

Now let us examine what this means when we use these new "blending" functions in the surface equation.

Certainly the surface will "contain" the four boundary curves, since the F functions still satisfy the four (primitive) stipulations. But now something else has been added.

Take, for example, the partial derivatives of the surface equation with respect to the variable u.

The derivative is:

$$uv_u = \begin{bmatrix} F_0'u & F_1'u \end{bmatrix} \begin{bmatrix} 0v \\ 1v \end{bmatrix} + \begin{bmatrix} u0_u & u1_u \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

$$- \begin{bmatrix} F_0'u & F_1'u \end{bmatrix} \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

This represents the tangent vector to the surface at any point, uv, in the u direction.

[The somewhat more orthodox notation is $\frac{\partial(uv)}{\partial u}$ which we write as uv_u.]

Now consider the tangent vectors along (across) a boundary. Choose, for example, the boundary Ov.

Then

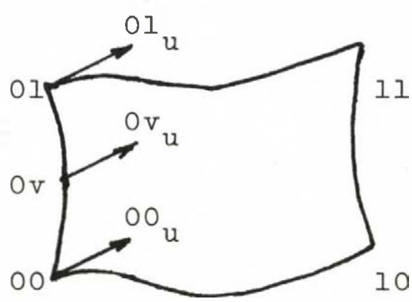
$$uv_u\Big|_{u=0} = \begin{bmatrix} F_0'0 & F_1'0 \end{bmatrix} \begin{bmatrix} 0v \\ 1v \end{bmatrix} + \begin{bmatrix} 00_u & 01_u \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

$$- \begin{bmatrix} F_0'0 & F_1'0 \end{bmatrix} \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}.$$

But the vector $\begin{bmatrix} F_0'0 & F_1'0 \end{bmatrix}$ is null, by virtue of the additional stipulations on the blending functions. Consequently, we learn that

$$uv_u\Big|_{u=0} = \begin{bmatrix} 00_u & 01_u \end{bmatrix} \begin{bmatrix} F_0 v \\ F_1 v \end{bmatrix}$$

Here again a sketch might be helpful:



The tangent vector along/across the boundary Ov is the "blended" vector function of the two terminal tangent vectors at 00 and at 01.

Now it is clear that if an adjacent "patch" has the same tangent vectors at 00 and at 01, with the same blending functions, then the two patches will join and will be tangent-vector-continuous at the boundary. This is to say that the two patches will be slope-continuous along/across their contiguous boundary. It is again striking that the tangent-vector regime (or "suite", as some call it) is entirely independent of the boundary curves themselves, except for the two tangent vectors 00_u and 01_u, which of course depend a little bit on the curves u0 and u1, but only when u = 0.

We can extend the stipulations on the nature of the F functions. We can have

$F_i j = \delta_{ij}$ which yields C^0 continuity.

$F_i' j = 0$ which yields C^1 continuity.

$F_i'' j = 0$ which yields C^2 continuity.

and so on.

When we define the F functions in this way, we can achieve C^n continuity between adjacent patches, quite automatically, without any necessity to pay any attention to details.

The F_0 function

$$F_0u = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -3 \\ 0 \\ 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} u^3 & u^2 \end{bmatrix} \begin{bmatrix} 2 \\ -3 \end{bmatrix} + 1 .$$

satisfies the conditions

$$F_0 0 = 1,$$
$$F_0 1 = 0,$$
$$F_0' 0 = 0,$$
$$F_0' 1 = 0.$$

The F_1 function is simply

$$F_1u = 1 - F_0u$$

The F_0 function

$$F_0u = \begin{bmatrix} u^5 & u^4 & u^3 \end{bmatrix} \begin{bmatrix} -6 \\ 15 \\ -10 \end{bmatrix} + 1$$

satisfies the conditions

$$F_0 0 = 1, \qquad F_0'' 0 = 0,$$
$$F_0 1 = 0, \qquad F_0'' 1 = 0,$$
$$F_0' 0 = 0,$$
$$F_0' 1 = 0.$$

Again:

$$F_1u = 1 - F_0u.$$

The surface equation can be written in a more compact quasi-indicial way:

$$(uv) = F_i u(iv) + F_j v(uj) - F_i u F_j v(ij).$$

i, j = 0, 1. We use the so-called Einstein convention in evaluating this expression.

Differentiate this expression twice with respect to u. The result is

$$(uv)_{uu} = F''_i u(iv) + F_j v(uj)_{uu} - F''_i u F_j v(ij)$$

Now set u = 0, for example. The F'' functions vanish.

$$(0v)_{uu} = F_j v(0j)_{uu} = F_0 v(00)_{uu} + F_1 v(01)_{uu}.$$

This shows that in a way analogous to the dependence of the first derivative vector regime on the two terminal tangent vectors, similarly in this case, when $F''_i j = 0$, the second derivative vectors along the boundary are determined by the values of these vectors at the ends of the 0v boundary. This permits us to obtain second derivative continuity where two patches join. We achieve curvature continuity automatically.

Unfortunately, there is a curious pathology connected with surfaces defined in this way. This fault is not serious in many cases, but it does sometimes appear and give unwanted results.

Take the cross-derivatives of the surface equation:

$$(uv)_{uv} = F_i'u(iv)_v + F_j'v(uj)_u - F_i'uF_j'v(ij)$$

Now when $(uv) = (ij)$, (at the corners,)

$$(ij)_{uv} = 0, \quad \text{a null vector.}$$

We sometimes call these four vectors the "corner twist vectors". In many cases this is relatively harmless, but in many other cases it exhibits itself as a kind of pseudo-flattening, when for example four adjacent patches have a corner in common.

We can remove this pathology by introducing another term into the surface equation. This term is

$$G_iu \;\; G_jv \; (ij)_{uv}.$$

Here we have introduced a new function, G, with the properties

$$G_{ij} = 0, \quad G_i'j = \delta_{ij}.$$

The augmented equation now reads

$$(uv) = F_iu(iv) + F_jv(uj) - F_iuF_jv(ij) + G_iuG_jv(ij)_{uv}$$

As before, we wish to see how the tangent vectors behave along a boundary. We differentiate with respect to u for example, and obtain

$$(uv)_u = F_i^!u(iv) + F_jv(uj)_u - F_i^!uF_jv(ij)$$

$$+ G_i^!uG_jv(ij)_{uv}$$

Now set $u = 0$, to be specific.

$$(0v)_u = F_jv(0j)_u + G_jv(0j)_{uv}.$$

This is the tangent vector function along the 0v boundary.

Written out, it is, in matric form,

$$(0v)_u = \begin{bmatrix} 00_u & 01_u \end{bmatrix} \begin{bmatrix} F_0v \\ F_1v \end{bmatrix}$$

$$+ \begin{bmatrix} 00_{uv} & 01_{uv} \end{bmatrix} \begin{bmatrix} G_0v \\ G_1v \end{bmatrix}$$

The surface equation now permits us to "design" the boundary curves, and interpolates a surface between them. The boundary tangent vectors and higher derivative vector functions along these boundaries are not, however, completely under our control.

We wish to specify, not only the boundary curves, but also the behavior of, to begin with, the tangent vector functions at the boundaries. The more general surface equation becomes

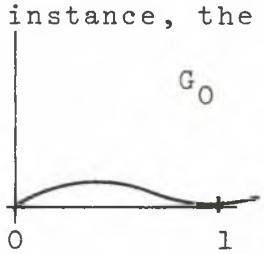
$$
\begin{aligned}
uv = {} & \begin{bmatrix} F_0u & F_1u & G_0u & G_1u \end{bmatrix} \begin{bmatrix} 0v \\ 1v \\ 0v_u \\ 1v_u \end{bmatrix} \\
& + \begin{bmatrix} u0 & u1 & u0_v & u1_v \end{bmatrix} \begin{bmatrix} F_0v \\ F_1v \\ G_0v \\ G_1v \end{bmatrix} \\
& - \begin{bmatrix} F_0u & F_1u & G_0u & G_1u \end{bmatrix} \left[\begin{array}{cc|cc} 00 & 01 & 00_v & 01_v \\ 10 & 11 & 10_v & 11_v \\ \hline 00_u & 01_u & 00_{uv} & 01_{uv} \\ 10_u & 11_u & 10_{uv} & 11_{uv} \end{array}\right] \begin{bmatrix} F_0v \\ F_1v \\ G_0v \\ G_1v \end{bmatrix}
\end{aligned}
$$

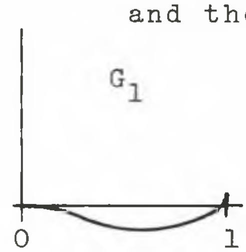
In this, typically, $0v_u$ is the desired vector function of v that describes the variation of the tangent vectors in the u direction, along the boundary. Again, this is an entirely arbitrary function of the independent variable.

The G functions, as mentioned before, satisfy the conditions

$$G_i j = 0 \qquad G_i' j = \delta_{ij} \qquad i, j = 0, 1$$

For instance, the G_0 function is and the G_1 is





(Observe that the G_1 function is everywhere negative from u = 0 to u = 1.)

Suitable functions that satisfy these conditions are the cubics

$$G_0u = \begin{bmatrix} u^3 & u^2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad \text{and} \quad G_1u = \begin{bmatrix} u^3 & u^2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The reader can easily verify that this equation does indeed yield a surface containing the four boundary curves, but now with the tangent-vectors at these boundaries under complete control.

In passing, we can point out that if the F and G functions are simply cubics, the vector

$$\begin{bmatrix} F_0u & F_1u & G_0u & G_1u \end{bmatrix} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

This matrix occurs in so-called Hermite interpolation, and also in "bi-cubic" surfaces, which are just a special case of the more general equation.

The G_0 function that satisfies the conditions

$$G_ij = 0 \quad G'_ij = \delta_{ij} \quad \underline{\text{and}} \quad G''_ij = 0$$

can be the quintic

$$G_0 u = \left[u^5 \quad u^4 \quad u^3 \right] \begin{bmatrix} -3 \\ 8 \\ -6 \end{bmatrix} + u$$

Similarly,

$$G_1 u = \left[u^5 \quad u^4 \quad u^3 \right] \begin{bmatrix} -3 \\ 7 \\ -4 \end{bmatrix}$$

In like fashion, we can introduce still another blending function, H. We can make a matrix to describe the stipulations in the F, G, and H functions:

$$\begin{bmatrix} F_i j & F'_i j & F''_i j \\ G_i j & G'_i j & G''_i j \\ H_i j & H'_i j & H''_i j \end{bmatrix} = \begin{bmatrix} \delta_{ij} & 0 & 0 \\ 0 & \delta_{ij} & 0 \\ 0 & 0 & \delta_{ij} \end{bmatrix}$$

with i, j = 0, 1, as usual.

Clearly we can extend such a definitive Kronecker delta matrix indefinitely, so as to make stipulations about even higher order blending functions, such as I, J and so forth.

The H functions make it possible to control curvatures across contiguous boundaries between patches, and make these curvatures continuous, and explicit.

The foregoing surface equation has been described by GORDON as a "Boolean-sum Surface". We define a Boolean-sum as follows:

$$A \oplus B = A + B - AB$$

We now introduce the Φ symbols, with their subscripts.
We write

$$\Phi = \Phi_1 \oplus \Phi_2$$

where the subscripts are intended to indicate which variables are held fixed.

For instance, Φ_1 means that in the surface equation, the first of the two variables uv is held fixed, while Φ_2 means that the second of the two variables is held fixed.

Now

$$= \Phi_1 \oplus \Phi_2 = \Phi_1 + \Phi_2 - \Phi_{12}.$$

Here the symbol Φ_{12} means that both variables are held fixed.

This is evidently descriptive of the structure of

$$uv = F_i u(iv) + F_j v(uj) - F_i u F_j v(ij)$$

where the first term has the first variable fixed, the second term has the second variable fixed, and the third term has both variables fixed.

But if we knew the form of

$$\Phi = \Phi_1 \oplus \Phi_2$$

we could write down the detailed surface equation, based on these substitutions. So the Boolean Φ equations really describe the form (but not the substance) of the surface equations.

Now consider the three-variable function (u v w). These variables control the position of a point that moves in its space of immersion with three degrees of freedom. One interpretation of such a locus is a three-dimensional "block", but there are other interpretations.

As is the case with surfaces, we will consider the functions that arise when we hold one of the three variables fixed and allow the other two to vary. These are (i v w), (u j w), (u v k). Here the indices i, j and k are allowed to take the values 0 and 1, so there are six such functions. Each represents a boundary surface, and the analogy with the six faces of a cube is evident.

We wish to construct the Boolean-sum structure of the locus. We use the Φ symbols:

$$\Phi = \Phi_1 \oplus \Phi_2 \oplus \Phi_3.$$

As before, the subscripts indicate which variables are held fixed.

We can now evaluate this Boolean-sum as follows:

$$\Phi = (\Phi_1 \oplus \Phi_2) + \Phi_3 - (\Phi_1 \oplus \Phi_2)\, \Phi_3$$

But
$$\Phi_1 \oplus \Phi_2 = \Phi_1 + \Phi_2 - \Phi_{12}.$$

Substituting,

$$\Phi = \Phi_1 + \Phi_2 - \Phi_{12} + \Phi_3 - (\Phi_1 + \Phi_2 - \Phi_{12})\, \Phi_3$$

$$= \Phi_1 + \Phi_2 + \Phi_3 - \Phi_{12} - \Phi_{13} - \Phi_{23} + \Phi_{123}.$$

This shows the <u>structure</u> of the point-locus equation. We can now write it out in detail:

$$u\ v\ w = F_i u(i\ v\ w) + F_j v(u\ j\ w) + F_k w(u\ v\ k)$$

$$- F_i u F_j v(i\ j\ w) - F_i u F_k w(i\ v\ k)$$

$$- F_j v F_k w(u\ j\ k)$$

$$+ F_i u F_j v F_k w(i\ j\ k).$$

We should be able to test this equation to see whether it "contains" one of the boundaries. Accordingly, let us set $u = 0$.

Then, since $F_0 0 = 1$, $F_1 0 = 0$, we obtain:

$$u\ v\ w\Big|_{u=0} = (o\ v\ w) + F_j v(o\ j\ w) + F_k w(o\ v\ k)$$

$$- F_j v(o\ j\ w) - F_k w(o\ v\ k)$$

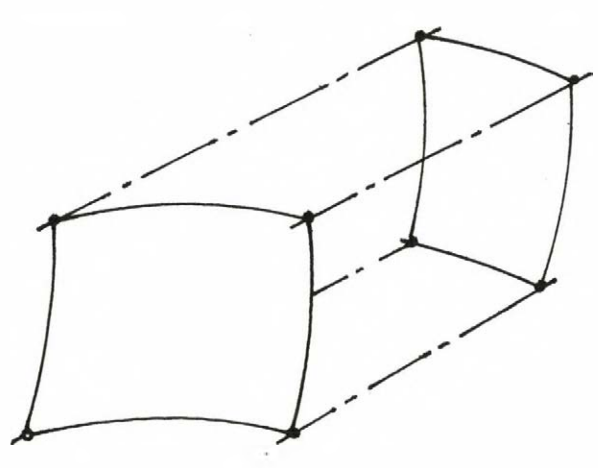
$$- F_j v F_k w(o\ j\ k)$$

$$+ F_j v F_k w(o\ j\ k).$$

This is just $o\ v\ w$ as we hoped it would be.

The six surfaces that bound this "block" are entirely arbitrary. That is to say, this point-locus is non-denumerable, in the same way that surfaces (or patches) are non-denumerable, since their boundaries are non-denumerable. It is a "transfinite" interpolation scheme.

Now consider a different problem. Let there be two surfaces and four curve "trajectories" of the corners of these surfaces.



Let the two surfaces be u v 0 and u v 1 and let the four trajectory curves be 0 0 w 0 1 w 1 0 w and 1 1 w.

Now the -structural equation is

$$\Phi = \Phi_3 \oplus \Phi_{12}$$

since the surfaces have only the third variable held fixed, while the curve trajectories have both their first and second variables held fixed. This Boolean-sum is

$$\Phi = \Phi_3 + \Phi_{12} - \Phi_{123}.$$

Now, using this expression as a guide, we write

$$(u\ v\ w) = F_k w(u\ v\ k) + F_i u F_j v(i\ j\ v)$$

$$- F_i u F_j v F_k w(i\ j\ k).$$

In even greater detail, this is

$$(u\ v\ w) = \underline{F_0 w(u\ v\ 0) + F_1 w(u\ v\ 1)}$$

$$+ F_0 u F_0 v(0\ 0\ w) + F_0 u F_1 v(0\ 1\ w)$$

$$+ \underline{F_1 u F_0 v(1\ 0\ w) + F_1 u F_1 v(1\ 1\ w)}$$

$$- F_0uF_0vF_0w(0\ 0\ 0) - F_0uF_0vF_1w(0\ 0\ 1)$$
$$- F_0uF_1vF_0w(0\ 1\ 0) - F_0uF_1vF_1w(0\ 1\ 1)$$
$$- F_1uF_0vF_0w(1\ 0\ 0) - F_1uF_0vF_1w(1\ 0\ 1)$$
$$- F_1uF_1vf_0w(1\ 1\ 0) - F_1uF_1vF_1w(1\ 1\ 1).$$

Notice that this equation makes specific reference to the two surfaces, the four curve trajectories, <u>and</u> to the <u>eight</u> corners of the locus. There are fourteen terms in the equation, so it is certainly not easy to evaluate, except by computer, of course.

Again it is possible to verify that the locus contains the two bounding surfaces and the four trajectories, but this is too tedious to récord here.

In a similar way, we could write the locus equation for a block defined by twelve curves, just as a cube is defined. The Boolean-sum would be

$$\Phi = \Phi_{12} \oplus \Phi_{13} \oplus \Phi_{23}.$$

As has been stressed before, the curves that bound and define surface patches can be any curves whatever, but there is a class of curves that are very convenient for their extreme simplicity and benign behavior. These are the so-called B-spline curves. In this discussion, we shall look only at "uniform, cubic B-splines". A point, moving along a B-spline sequence, may be described by the vector-valued equation

$$p = \frac{1}{6}\begin{bmatrix} s^3 & s^2 & s & 1\end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} v_i \\ v_{i+1} \\ v_{i+2} \\ v_{i+3}\end{bmatrix}$$

where the dummy variable s is

$$s = FR(n\ u)$$

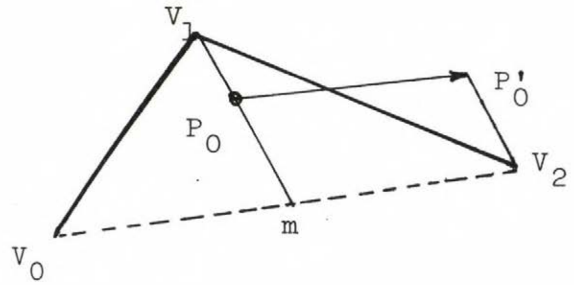
the fractional part of $n\ u$, and the index i is

$$i = INT(n\ u)$$

the integer part of $n\ u$.

n refers to the number of cubic segments in the sequence of segments, u is the proper independent variable, $u \in (0, 1)$ and the v_i are the vertices of an open or closed polygon that controls the shape of the curve.

Consider three vertices of such a polygon, say v_0 v_1 and v_2.



When we substitute $s = 0$ in the equation, we find that these three polygonal vertices define the position of the point p_0. It turns out to lie on the median $(v_1 m)$ of the triangle of points, and one-third of the distance from v_1 to m. This is a simple geometric fact about cubic B-splines that turns out to be very useful.

Furthermore, the tangent vector to the spline at p_0 (which may be thought of as a velocity vector) is parallel to $v_0 v_2$ and has a magnitude equal to half this distance, or specifically,

$$p_0' = \frac{v_2 - v_0}{2}$$

The second derivative vector at p_0 also has very simple geometric properties. It is directed along the median of the triangle, with a magnitude of twice the length of the median.

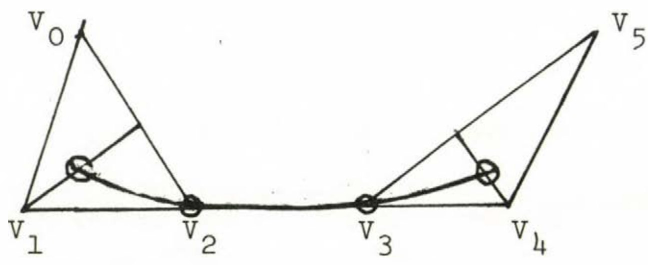
Vectorially, we can write

$$p_0'' = v_0 - 2v_1 + v_2.$$

This shows that the curvature of the spline at p_0 is also completely defined by the three vertices.

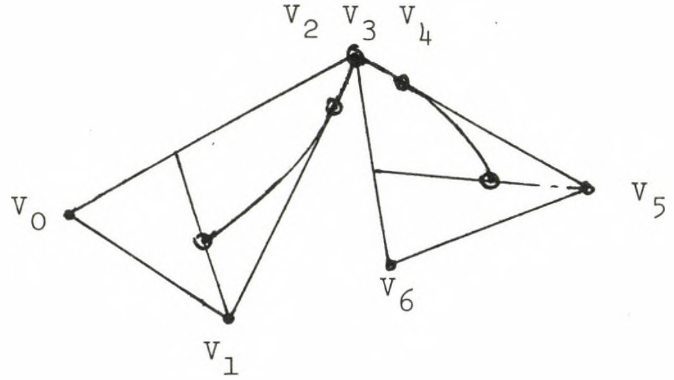
From these interesting and simple geometric relationships, we can deduce that when <u>four</u> or more vertices are collinear, the spline that they generate is the line of collineation.

Consider for a moment the vertices



They generate a curve consisting of three cubic segments (whose ends are indicated by small circles) which are C^2 continuous where they join, and of which the middle segment is straight.

Now consider the polygon in the following sketch, where three polygonal vertices coincide:

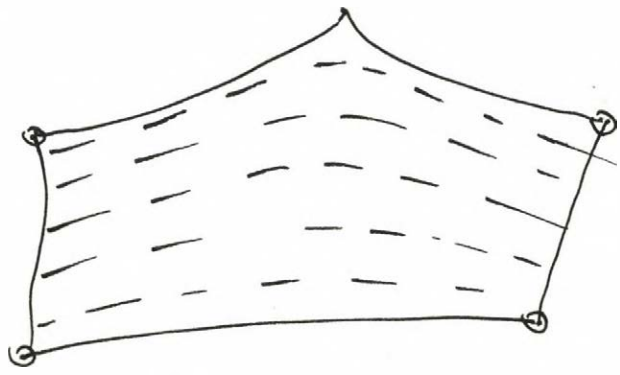


The seven vertices of this polygon generate 7 - 3 = 4 spline segments, of which, as the sketch indicates, two are short straight segments, entering and leaving the triple vertex $v_2 \ v_3 \ v_4$.

The curve has a cusp at $v_2 \ v_3 \ v_4$. But what is striking and seemingly paradoxical, the curve is C^2 continuous there!

To anyone accustomed to thinking of scalar functions of a variable, this is shocking. But in the vector-valued curve space, there is really no paradox at all. The three coincident vertices form a triangle, (a degenerate one, to be sure) and within this triangle is a point at which both the tangent-vector and the second-derivative vector are zero. But they <u>get</u> to be zero in a smooth continuous way, not in a jump-discontinuous way. Consequently, the simple <u>slope</u> of the branches of the cusp are different, and discontinuous, while the vector-valued function is C^2 continuous.

With curves that have these extraordinary properties, we can construct surfaces with four boundaries,



as the sketch suggests.

and with one of the boundary curves cuspidal, so that the surface <u>seems</u> to have five distinct boundary curves. Neverthe-less, the interior of the bounded surface is everywhere C^2 "smooth". This is, of course, a remarkable result.

BEZIER CURVES AND SURFACES,

WITH APPLICATIONS TO BLENDING FUNCTIONS

by

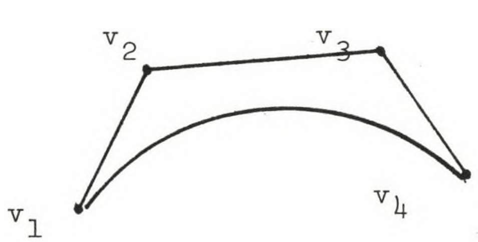
S. A. COONS

Computer and Automation Institute

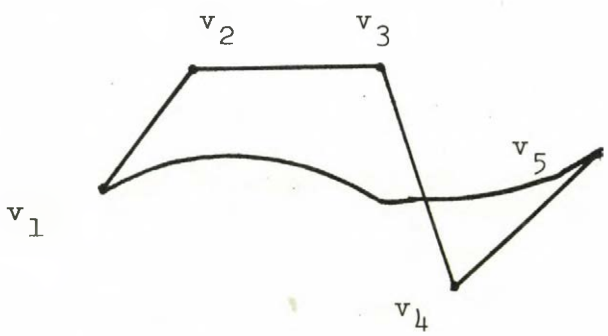
Hungarian Academy of Sciences

The following is a review of Bèzier curve and surface methods, as modified by FORREST.

Bèzier introduces the notion of a curve generated in such a way as to <u>interpolate</u> the slopes and vertices at the two ends of an "open" polygon, and to <u>approximate</u> the intermediate vertices of the polygon. Thus four vertices yield a curve:



Similarly, five vertices yield a curve:



and indeed it will be seen that the polygon can have as many vertices as we wish.

We choose polynomials as the (vector) curve functions, and in particular these polynomials are known as "Bernstein polynomials". For obvious algebraic reasons, the <u>order</u> of the polynomial is the same as the <u>number</u> of vertices.

[The <u>degree</u> of the polynomial is one less than the <u>order</u> of the polynomial.]

Thus the <u>degree</u> of the polynomial is the same as the <u>number of sides</u> of the defining polygon.

Qualitatively, these Bèzier curves imitate the shape of the defining polygon.

For four vertices, the vector point function p that describes the curve is given by

$$p = \left[(1-u)^3 \mid 3(1-u)^2 u \mid 3(1-u)u^2 \mid u^3 \right] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

where u is the independent variable.

For n+1 vertices, the vector function is

$$p = \left[\binom{n}{n}(1-u)^n \mid \binom{n}{n-1}(1-u)^{n-1}u \mid \binom{n}{n-2}(1-u)^{n-2}u^2 \mid \quad \ldots \right.$$

$$\left. \ldots \binom{n}{0} u^n \right] \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix}$$

- . -

Consider the four-vertex case. The basis vector

$$\left[(1-u)^3 \mid 3(1-u)^2 u \mid 3(1-u)u^2 \mid u^3 \right]$$ can be

rewritten in matric form as

$$\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Thus we have isolated the "primitive polynomial basis" $\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$ from all constant coefficients that will appear as post-multipliers, and, as we shall see, this is a great convenience.

The square matrix has two significant characteristics:

1. It is a symmetric matrix.
2. Its top-row consists of the binomial coefficients of $(1-u)^3$, which is the same as saying that these numbers appear in the PASCAL TRIANGLE, (in the fourth column,)

$$\begin{array}{ccccc} 1 & 1 & 1 & \boxed{\begin{array}{c}1\\3\\3\\1\end{array}} & \begin{array}{c}1\\4\\6\\4\\1\end{array} \\ & 1 & 2 & & \\ & & 1 & & \end{array} \qquad \text{etc. -}$$

but they appear with alternating signs.

The columns of the Bèzier matrix can be filled in by multiplying the columns of the Pascal triangle by the numbers in the top row, and observing the alternating sign requirement.

These qualitative remarks apply to all orders of the Bèzier matrix. Thus we can have

$$\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \text{ and for fifth order,}$$

$$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

[Later on, we shall have occasion to use the sixth order Bèzier matrix, which we can write by inspection.]

- . -

Now, for illustrative purposes, let us compute for the cubic, the point-vector at $u = 0$ and at $u = 1$, and also the first derivative vectors (the velocities) of the moving point at these values of u.

The primitive basis $\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$ takes the values

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \begin{matrix} \text{at } u = 0, \\ \text{at } u = 1, \end{matrix}$$

and the first derivative, $\begin{bmatrix} 3u^3 & 2u & 1 & 0 \end{bmatrix}$ takes the values

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}.$$

We have

$$\left[\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{array}\right] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

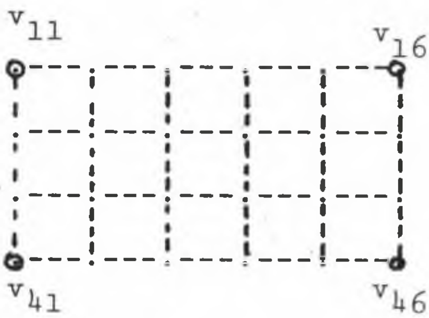
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

We now can write, after performing the multiplication:

$$\left[\begin{array}{c} p(0) \\ p(1) \\ \hline p'(0) \\ p'(1) \end{array}\right] = \left[\begin{array}{c} v_1 \\ v_4 \\ \hline 3(v_2-v_1) \\ 3(v_4-v_3) \end{array}\right]$$

which verifies that the (cubic) polynomial does indeed interpolate the end-points and end-slopes of the defining polygon. This is a very nice property, shared by all orders of Bèzier curves.

BEZIER SURFACES.



Now consider the array of vertices v_{ij} which we have sketched as a simple 4 x 6 rectangular array, but which ordinarily is not rectangular at all, since, for example, the top row of vertices $v_{11}\ v_{12}\ v_{13}\ v_{14}\ v_{15}\ v_{16}$ really constitute a Bèzier polygon, and define a curve, in general. We will use the symbols B_4 and B_6 to represent the fourt-order and sixth-order Bèzier matrices, and write

$$p = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} B_4 \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} & v_{16} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} & v_{26} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} & v_{36} \\ v_{41} & v_{42} & v_{43} & v_{44} & v_{45} & v_{46} \end{bmatrix} B_6 \begin{bmatrix} w^5 \\ w^4 \\ w^3 \\ w^2 \\ w \\ 1 \end{bmatrix}$$

This is a vector function of <u>two</u> independent variables u and w, and consequently describes the locus of the point p that moves with two degrees of freedom: in other words: it moves on a surface.

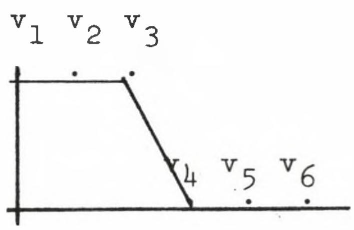
[Strictly speaking, we should write B_6^T, the transpose of B_6, but since B_6 is symmetric, we ignore the transpose sign.]

The surface <u>interpolates</u> the four corner vertices v_{11} v_{16} v_{41} v_{46} and is tangent to the eight corner slopes, in strict analogy to the curve cases. Internally, the surface <u>approximates</u> all the other vertices. It requires 24 vector-valued vertices to describe the surface, and eight of these vertices are arbitrary "internal vertices".

[Since presumably the "space of immersion" of this surface is 3-dimensional, each of the v_{ij} consists of three coordinates, so there are 3 x 24 = 72 numbers required, in all.]

Such surfaces are called "tensor-product" surfaces.

SOME APPLICATIONS OF BEZIER CURVE METHODS.



Consider the polygon with six vertices, where the v_i are scalar valued (which is to say that they are vector-valued, but that these vectors have only one component.) Then

$$p = \begin{bmatrix} u^5 & u^4 & u^3 & u^2 & u & 1 \end{bmatrix} B_6 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$ if we let

v_1 v_2 $v_3 = 1$ and v_4 v_5 $v_6 = 0$.

The B_6 matrix is

$$\begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \\ 5 & -20 & 30 & -20 & 5 & 0 \\ -10 & 30 & -30 & 10 & 0 & 0 \\ 10 & -20 & 10 & 0 & 0 & 0 \\ -5 & 5 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which we write by inspection, as has already been described.

Then, multiplying out, we get

$$p = \begin{bmatrix} u^5 & u^4 & u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -6 \\ 15 \\ -10 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

which can be simplified to read

$$p = \begin{bmatrix} u^5 & u^4 & u^3 \end{bmatrix} \begin{bmatrix} -6 \\ 15 \\ -10 \end{bmatrix} + 1.$$

Let us now compute the function and derivatives:

$$\begin{bmatrix} p(0) \\ p'(0) \\ p''(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

which says that p passes through the point $v_1 = 1$, with zero first and second derivatives there. Again,

$$\begin{bmatrix} p(1) \\ p'(1) \\ p''(1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 5 & 4 & 3 \\ 20 & 12 & 6 \end{bmatrix} \begin{bmatrix} -6 \\ 15 \\ -10 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

This function, obtained at very little expense, is evidently the same as the so-called blending function F_0, with its requirements that

$$F_0 0 = 1 \qquad F_0 1 = 0$$
$$F_0' 0 = 0 \qquad F_0' 1 = 0$$
$$F_0'' 0 = 0 \qquad F_0'' 1 = 0 \quad .$$

- . -

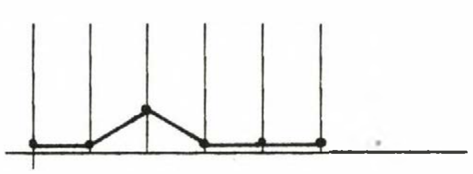
In a similar way, and very easily, we can obtain the quintics for F_1 G_0 G_1 H_0 H_1 mentioned and used elsewhere.

- . -

Indeed, let us construct the H_0 function, that obeys the constraints

$$H_0 0 = 0 \qquad H_0 1 = 0$$
$$H_0' 0 = 0 \qquad H_0' 1 = 0$$
$$H_0'' 0 = 1 \qquad H_0'' 1 = 0 \quad .$$

By naive algebraic methods, this would require the inversion of a 6 x 6 matrix. But we construct a Bèzier polygon:



with vertices $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

and find at once

$$p = \begin{bmatrix} u^5 & u^4 & u^3 & u^2 \end{bmatrix} \begin{bmatrix} -10 \\ 30 \\ -30 \\ 10 \end{bmatrix}$$

This is indeed the form we seek, except possibly for a constant factor. Compute

$$\begin{bmatrix} p_0(0) \\ p_0'(0) \\ p_0''(0) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -10 \\ 30 \\ -30 \\ 10 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}$$

Evidently then we <u>should</u> write

$$p = \begin{bmatrix} u^5 & u^4 & u^3 & u^2 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} \qquad \text{for } H_0.$$

- . -

Elsewhere we have the need for a LINEAR QUINTIC FUNCTION, that has the property of passing through the two points p_0 and p_1 and with zero velocity and zero acceleration vectors at these points.

The Bèzier polygon is:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \hline v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} p_0 \\ p_0 \\ p_0 \\ \hline p_1 \\ p_1 \\ p_1 \end{bmatrix}$$

and the quintic that results is, after a little rearrangement:

$$p = \begin{bmatrix} u^5 & u^4 & u^3 \end{bmatrix} \begin{bmatrix} 6 \\ -15 \\ 10 \end{bmatrix} \begin{bmatrix} p_1 - p_0 \end{bmatrix} + p_0 .$$

The quantity

$$\begin{bmatrix} u^5 & u^4 & u^3 \end{bmatrix} \begin{bmatrix} 6 \\ -15 \\ 10 \end{bmatrix}$$ can be regarded as a new

parameter, say Φ, and evidently

$$\Phi \begin{bmatrix} p_1 - p_0 \end{bmatrix} + p_0$$ is linear in Φ,

although non-linear in u.

Hence the name, "Linear quintic function".

It is used in the modification of piece-wise curve shapes, as cescribed elsewhere.

REMARKS ABOUT B-SPLINES,

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

The writer has a prejudice against what are known as "NON-UNIFORM" B-splines, as distinguished from "UNIFORM" B-splines. This prejudice arises from two considerations:

1. UNIFORM splines are very much simpler to compute; they depend upon a single basis matrix, unchanging for some chosen polynomial order, while the non-uniform splines depend upon basis matrices (or their equivalent) that are different for each case of non-uniformity, and must consequently be recomputed.

2. It turns out that, except for extraordinarily pathological cases, the curves obtained from non-uniform splines are not substantially different, qualitatively, from those obtained from the uniform case. (See Reisenfeld.) So the additional computational complexity seems scarcely worthwhile.

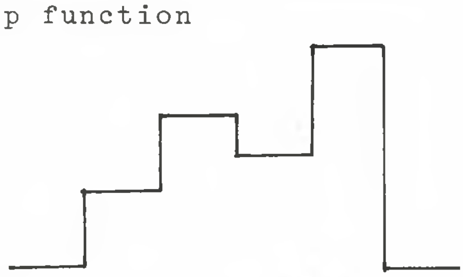
The Uniform B-spline basis

There are (at least) four ways to construct a basis matrix for uniform B-splines.

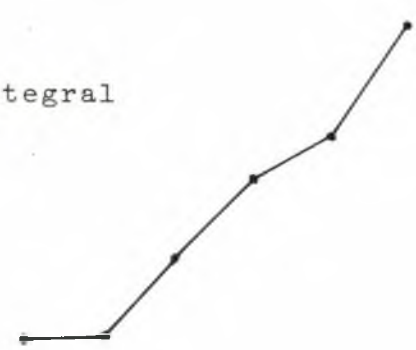
The (possibly) most sophisticated, and certainly the most general way, since it embraces non-uniform spline bases as well as uniform bases, is the algorithm of COX and DEBOOR. The details of this algorithm can be found elsewhere. It has appeared in the literature of several authors, notably Riesenfeld and Gordon and is well known.

Another and probably closely related way is by means of integration of polynomials.

We can think of the integral operator as a "smoothing" operator. The step function



has an integral



which is sometimes called a "ramp function".

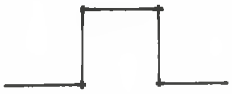
The step-function has finite discontinuities (it is C^{-1} continuous, we say).

The ramp function is C^{0} continuous, we say, meaning that each piece-wise (linear) interval "connects" with its neighbours, so that the result is single-valued even at these junctions.

If in turn we integrate this ramp-function, we obtain a sequence of (parabolic) segments that are slope-continuous, or first-derivative continuous, at the junctions. We say that it is C^{1} continuous. This is just a way of pointing out that the ramp function describes the slope of the parabola-function, and is itself continuous, so there is no discontinuity in the slope of its integral.

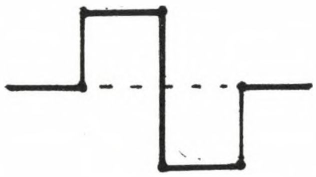
And so forth.

The simplest B-spline basis function is the single step:

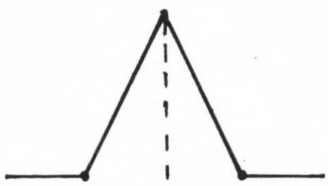


Later on, we will describe the significance of this basis. It yields, from a sequence of vertices V_1 V_2 V_3 ... V_i just the vertices themselves, and we can call it the "punctate" curve. It does <u>not</u> yield the lines joining these points, but only the points themselves.

Now, form the step function



by translating the single step to the right and reversing its sign. Integrate this, and we obtain the "tent function"

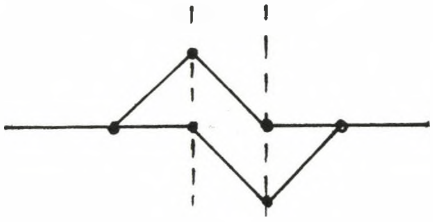


This function can be described by the matrix

$$\begin{bmatrix} S & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

where the two columns of the matrix describe, piece-wise, the two-sides of the tent, and S is the independent variable.

Now, as before, translate to the right and change sign. We obtain



This function can be described by the sum of the two constituents

$$\begin{bmatrix} s & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} s & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} s & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

We now propose to integrate this compound function. Its integral is

$$\begin{bmatrix} \frac{s^2}{2} & s \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

We leave the question of constants of integration until later.

Rewrite the leading vector:

$$\begin{bmatrix} \frac{s^2}{2} & s \end{bmatrix} = \frac{1}{2} \begin{bmatrix} s^2 & s \end{bmatrix} \begin{bmatrix} 1 & \\ & 2 \end{bmatrix},$$ and call the

square (diagonal) matrix the "integrator matrix".

Then $\begin{bmatrix} 1 & \\ & 2 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \end{bmatrix}$

Now introduce the constants of integration, writing

$$\begin{bmatrix} s^2 & s & 1 \end{bmatrix} \left[\begin{array}{ccc} 1 & -2 & 1 \\ 0 & 2 & -2 \\ \hline 0 & c_1 & c_2 \end{array}\right]$$

To satisfy continuity, the C_1 must be the value achieved by the first segment when S = 1, which gives $C_1 = 1$, similarly, C_2 must be the value of the integral at the end of the second segment, or $C_2 = -2 + 2 + 1 = 1$.

The integral matrix is, with its constants of integration,

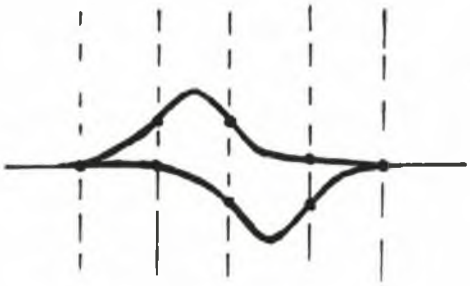
$$\begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

and the resulting basis function is

$$\frac{1}{2} \begin{bmatrix} S^2 & S & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

It consists of three parabolic segments, C^1 continuous where they join.

We now take this result, translate, reverse sign, and obtain



The (segment) matrices are

$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & -1 & 2 & -1 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & -1 & -1 \end{bmatrix} \text{ and}$$

their sum is the matrix $\begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 2 & -4 & 2 \\ 0 & 1 & 0 & -1 \end{bmatrix}$

Again we propose to integrate the piecewise function that this matrix generates.

The integral of its basis vector $\begin{bmatrix} s^2 & s & 1 \end{bmatrix}$ is the new basis $\begin{bmatrix} \frac{s^3}{3} & \frac{s^2}{2} & s \end{bmatrix}$ which we rewrite as

$$\frac{1}{6} \begin{bmatrix} s^3 & s^2 & s \end{bmatrix} \begin{bmatrix} 2 & & \\ & 3 & \\ & & 6 \end{bmatrix}$$

Now the integral matrix is the product

$$\begin{bmatrix} 2 & & \\ & 3 & \\ & & 6 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 2 & -4 & 2 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -6 & 6 & -2 \\ 0 & 6 & -12 & 6 \\ 0 & 6 & 0 & -6 \end{bmatrix}$$

When we supply the constants of integration, this becomes

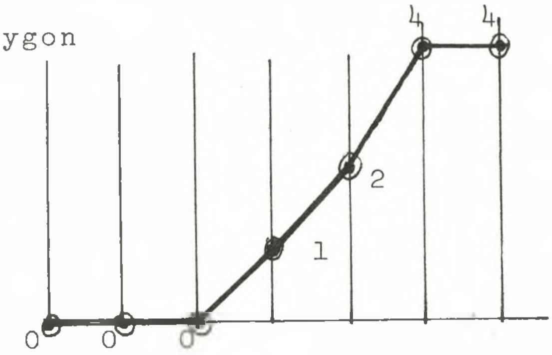
$$\begin{bmatrix} 2 & -6 & 6 & -2 \\ 0 & 6 & -12 & 6 \\ 0 & 6 & 0 & -6 \\ \hline 0 & 2 & 8 & 2 \end{bmatrix}$$

Recalling the factor of $\frac{1}{2}$ from the first integration, this yields, ultimately, the cubics describing the four segments of the, (by now,) C^2 continuous basis function,

$$\frac{1}{6}\begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 3 & 0 & -3 \\ 0 & 1 & 4 & 1 \end{bmatrix}$$

To anyone familiar with uniform cubic B-splines, this is an expected result.

As a third scheme, we can use Bèzier polynomials to construct the cubic (or any) B-spline basis matrix. Consider the Bèzier polygon



and similarly, symmetrically, for the remaining two cubic segments.

The two segments shown have vertices

$$\begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 0 & 4 \\ 1 & 4 \end{bmatrix}$$ and we can invoke right-left symmetry.

The Bèzier curves are given by the matrix product

$$\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \left[\begin{array}{cc|cc} 0 & 1 & 4 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 4 & 2 & 0 \\ 1 & 4 & 1 & 0 \end{array}\right]$$

$$= \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 3 & 0 & -3 \\ 0 & 1 & 4 & 1 \end{bmatrix} \quad \text{as before}$$

Such a scheme is attractive from the standpoint of directness and simplicity, provided we know how to select the Bèzier vertices. But this is the difficulty. It is not insurmountable, but it is tricky.

The fourth scheme turns out also to be fairly direct and simple. It depends upon the naive notion of imposing continuity conditions between the four cubic segments, and finding a solution for the resulting matrix equations.

We seek a matrix such that the four segments are given by

$$\begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}$$

Now the expression

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 2 & 1 & 0 \\ 6 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \left[\begin{array}{c|ccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{array}\right] \begin{bmatrix} a_{i+1} \\ b_{i+1} \\ c_{i+1} \\ d_{i+1} \end{bmatrix}$$

describes the C^0 C^1 and C^2 continuity requirements between the i segment and the $i+1$ segment. We can invert the right-hand square partition of the matrix on the right, and obtain, after multiplication,

$$\begin{bmatrix} 3 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} b_{i+1} \\ c_{i+1} \\ d_{i+1} \end{bmatrix}$$

The quantity a_{i+1} does not appear here, but provided we know the a_i quantities, we can obtain very simply the $\begin{bmatrix} b_{i+1} \\ c_{i+1} \\ d_{i+1} \end{bmatrix}$

It turns out that the a_i quantities that appear in the top row of the matrix are solutions of a set of linear homogeneous equations, and since the matrix involved happens to be singular, there is a non-trivial solution (or set of solutions).

In the present cubic case, it turns out that

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 3 & -1 \end{bmatrix}$$

is <u>a</u> solution for the a_i.

This leads to a simple recursive procedure: since we can <u>supply</u> the a_i, we can <u>compute</u> the b_{i+1} c_{i+1} d_{i+1}. But then again we <u>supply</u> the a_{i+1}, and compute the next set of b_{i+2} c_{i+2} d_{i+2}, and so forth.

Notice the ubiquitous appearance of the elements in the PASCAL TRIANGLE. The a_i's are the elements $\begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix}$ with alternating sign. The matrix

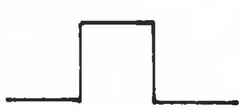
$$\begin{bmatrix} 3 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

is the Pascal triangle, except for the ommission of the 1 in the first column.

Similar remarks apply for all orders of B-spline basis matrices. For the quartic case, the a_i are just $\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}$, and the recursion matrix is

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ 6 & 3 & 1 & 0 & 0 \\ 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

both of which can be written down by inspection.
In the case of the most primitive of all B-spline bases, the variable S does not appear at all. Points are described by



$$p = \begin{bmatrix} 1 \end{bmatrix} V_i$$

where the bracketed "one" is the basis matrix. There is no curve, but only the point vertices V_i. The tent function, the next higher order, gives

$$p = \begin{bmatrix} S & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V_i \\ V_{i+1} \end{bmatrix}$$

and this describes the sides of the $V_1\ V_2\ V_3 \ldots$ polygon.

THE MODIFICATION OF THE SHAPE OF PIECE-WISE CURVES

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

We have, let us suppose, a sequence of piece-wise curves, K_1, K_2, K_3 ... that taken together describe some shape. Let it also be supposed that these curve pieces (or segments) are C^2 continuous at their joints, or knots.

Such curve segments can be of any description whatever. They might be the segments of a cubic B-spline sequence, for example.

We wish to modfy these K_i so as to change the shape they describe, while still preserving the C^2 continuity condition.

We introduce what we will call the "LINEAR QUINTIC FUNCTION" which we define as

$$L = [\ S^5\ S^4\ S^3\ ] \begin{bmatrix} 6 \\ -15 \\ 10 \end{bmatrix} [\ P_1 - P_0\ ] + P_0.$$

Here S is the independent variable, S $\in$ (0, 1) and P_0 and P_1 are vector constants at S=0 and S=1 respectively.

This function has the following properties:

$$L(0) = P_0 \qquad L'(0) = 0 \qquad L''(0) = 0$$

$$L(1) = P_1 \qquad L'(1) = 0 \qquad L''(1) = 0.$$

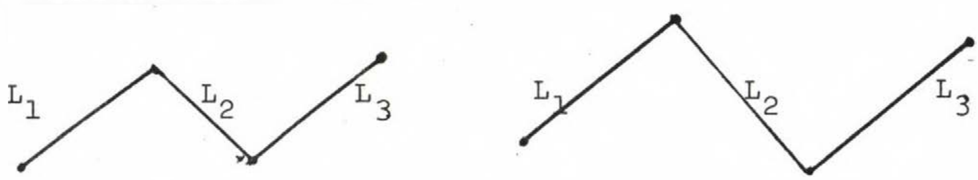
Furthermore, it is asy to see that if we consider the scalar

$[\, S^5 \; S^4 \; S^3 \,] \begin{bmatrix} 6 \\ -15 \\ 10 \end{bmatrix}$ as a variable <u>coefficient</u>,

L is linear. All point vectors generated by L lie on the line from P_0 to P_1. (Hence the name, LINEAR QUINTIC.)

The first and second derivatives (vectors) vanish (or are null-vectors) at S = 0 and at S = 1. We can call the first-derivative vector the "velocity" of the moving point, and the second derivative vector the "accelaration" of the moving point, with respect to S.

Now it is appropriate to consider a sequence of such linear quintics, L_1 L_2 L_3 ... .



The resulting curve is a polygon, ostensibly, with abrupt slope discontinuities at the joints. But paradoxically, the curve is velocity and acceleration, (or C_1^2) continuous at these junctions, when we consider these vectors as functions of the independent, or "driving" variable S.

The driven point undergoes a (negative) acceleration as it approaches its terminus. Simultaneously the acceleration approaches zero as well. So both velocity and acceleration smoothly approach zero, and then, on the next segment, both increase from zero smoothly and continuously.

Now, returning to the vector curve functions K_1 K_2 K_3, construct the compound vector function

$$Q = (1 - \alpha) + \alpha L.$$

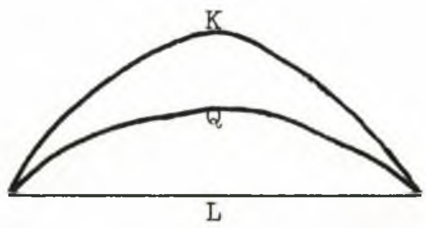
Here α is some arbitrarily chosen scalar constant. Since both K and L are C^2 continuous where they join their neighbours, the new vector function Q is likewise C^2 continuous at these junctions.

When $\alpha = 0$, $Q = K$,

When $\alpha = 1$, $Q = L$,

with other shapes in between (and beyond, when $\alpha<0$ or $\alpha>1$.)

Typically,



for some choice of α.

The vector function Q has velocity and acceleration vectors with directions the same as those of K but with altered magnitudes, according to the choice of the scalar α .

Some authors associate some such quantity as α with the "tension" on the spline sequence, or curve sequence. But the choice of linear combinations of K and L to yield Q is entirely arbitrary, and really has nothing to do with "tension", except in the case when Q and L are "almost" the same (the "linearized approximation" to true splines).

Nevertheless, the resulting family of curve shapes are interesting, simple to compute, andwell-behaved.

Q is, of course, usually a quintic.* But it has only K (already established) and L, (always the same quintic form) and an arbitrarily chosen quintity α , as constrains. So,

---

* When K is a polynomial of not higher than fifth degree. But K can be <u>any</u> function, satisfying C^2 continuity conditions.

even though it is quintic, it is not necessary to specify the <u>six</u> constraints that ordinarily would be required.

Evidently this idea can be extended to yield modifications that have C^3 ... C^4 continuity curve sequences.

VARYING "TENSION"

At first glance, it might seem that the quantity α must be the same for every segment Q. But paradoxically (so it seems) the number α can itself vary as some function of the independent piece-wise variable S. The only requirement, so as not to disturb the C^n continuity of the compound curve Q at the joints, is that the α quantity be at least C^{n-1} continuous at these joints. The proof follows.

We have the K-curve modifications,

$Q = K(1-\alpha) + L\alpha$ the piecewise curve Q in "pseudo-tension".

Rewrite this as the equivalent form,

$$Q = K + (L-K)\,\alpha \,.$$

Its first derivative is

$$Q' = K' + (L' - K')\,\alpha + (L - K)\,\alpha' .$$

At a <u>terminal</u> of the piece-wise segment, L-K =0, because the "linear polynomial" L is just the chord joining the terminals of K.

But also, at a terminal, L' = 0 because of the choice of the L function.

So at a terminal, making appropriate substitutions,

$$Q' = K' - K'\alpha \; .$$

This says that two curve segments, Q_1 and Q_2 will be continuous, C^1, at their common terminals, (their junctions) in case α is C^0 continuous there.

Set, for notational simplicity, $R = L - K$.
R is just a compound function, used temporarily.

We are now on the way to describe C^2 continuity at junctions.

We have already, from $Q = K + R\alpha$,

$$Q' = K' + R'\alpha + R\alpha'$$

and differentiating again,

$$Q'' = K'' + R''\alpha + R'\alpha' + R'\alpha' + R\alpha''$$

$$= K'' + R''\alpha + 2R'\alpha' + R\alpha''.$$

Now at a <u>junction</u> (but not elsewhere)

$R = 0$ since $K = L$ there.

$R'' = L'' - K''$

$= -K''$ since $L''=0$ there.

$R' = L' - K'$

$= -K'$ since $L'=0$ there.

Hence

$$Q'' = K'' - K''\alpha \; - 2K'\alpha'.$$

Accordingly, two Q segments will be C^2 continuous at a junction in case

$$\left.\begin{array}{l} L = K \\ L' = 0 \\ L'' = 0 \end{array}\right\}$$ and K is C^2 continuous at

the junction (which is to be supposed) <u>and</u> the "pseudo-tension" parameter α is C^1 continuous there. (Observe the weaker restriction on the α function.)

Again,

$$Q''' = K''' + R'''\alpha \; + 3R''\alpha' + 3R'\alpha'' \; + R\alpha'''.$$

If we have made L a "linear-polynomial function", then R= 0, at a terminal,

$$R''' = -K'''$$
$$R'' = -K''$$
$$R' = -K'$$ which gives

$$Q''' = K''' - K'''\alpha \; - 3K''\alpha' - 3K'\alpha.$$

Evidently if K''', K'' and K' are continuous with their neighbours, and if $\alpha_1\alpha'$ and α'' are likewise continuous with <u>their</u> neighbours, the compound <u>induced</u> functions Q will likewise be continuous, C''', with <u>their</u> neighbours.

Since, the "pseudo-tension" function α can take many forms, (it is piece-wise) it may be expedient to make it a function such that, similarly to the L function (the "linear polynomial function") it takes zero first, second, and higher derivatives at its terminals, as required.

For example, the α function that is C^0 continuous at the junctions is simply

$$\alpha = [\ u\ ]\ [\ \alpha_1 - \alpha_0\ ] + \alpha_0, \quad \text{(a linear form).}$$

This is simply linear from α_0 to α_1.

The α function that is C' continuous at the junctions can be

$$\alpha = [\ u^3\ u^2\ ] \begin{bmatrix} -2 \\ 3 \end{bmatrix} [\ \alpha_1 - \alpha_0] + \alpha_0.$$

This of course is cubic between α_0 and α_1, and it ensures C' continuity with its neighbours. And so forth.

All of the foregoing describes a scheme for modification of the segments of a piece-wise continuous sequence of segments such that we may introduce a "pseudo-tension parameter" α, segment by segment, and vary α so as to give varying tensions, (or average tensions) in each segment.

It is somewhat like saying that the "spline" experiences some <u>tangential</u> forces (influences) along its length. Hence the tension (or pseudo-tension) could be expected to vary. It is just in imitation of this effect that the foregoing (very simple) mathematics has been constructed.

CLINE, A.K.:

"SCALAR - AND PLANAR-VALUED CURVE FITTING USING SPLINES UNDER TENSION"
Comm. ACM 17, 4, 1974. pp. 218-220.

NIELSON, G.M.:

"SOME PIECE-WISE POLYNOMIAL ALTERNATIVES TO SPLINES UNDER TENSION"
COMPUTER-AIDED GEOMETRIC DESIGN -
ACADEMIC PRESS - NEW YORK - 1974.

SCHWEIKERT, D.G.:

"AN INTERPOLATION CURVE USING A SPLINE IN TENSION"
J. MATH. PHYSICS 45,
1966, pp. 312-317.

PHOTOGRAMMETRIC MEASUREMENT.

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

The following was developed first by BURTON, then a graduate student at the University of Utah. It is possible that SUTHERLAND suggested the method to BURTON.

It depends upon the matric algebra of projective transformations.

For this discussion, assume that there are two cameras focussed on the $[x\ y\ z]$ space to be measured. The placement of the cameras is not important, nor need their optical characteristics be in any way matched. (The two cameras can be quite different.)

Each camera produces a two-dimensional picture of the $[x\ y\ z]$ object space. Let the coordinates of points in one picture be $[u\ v]$ and in the other picture let them be $[s\ t]$.

We use homogenous coordinates, so that points in the object space have the <u>four</u> coordinates $\begin{bmatrix} x & y & z & 1 \end{bmatrix}$ and points in the images have <u>three</u> coordinates, $\begin{bmatrix} hu & hv & h \end{bmatrix}$ and $\begin{bmatrix} ks & kt & k \end{bmatrix}$ where h and k are unknown.

Then the photographic transformation is

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} = \begin{bmatrix} hu & hv & h \end{bmatrix}$$

with a_{ij} and h unknown.

Typically, in this, the first column gives

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} = hu,$$

but also the third column gives h, so that

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix} u = hu,$$

or

$$\begin{bmatrix} ux & uy & uz & u \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix}$$

or

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} - \begin{bmatrix} ux & uy & uz \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = a_{43}\, u.$$

We can now write, placing the a_{ij} in a column,

$$\left[\begin{array}{cccc|cccc|ccc} x & y & u & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz \end{array}\right] \left[\begin{array}{c} a_{11} \\ 21 \\ 31 \\ 41 \\ \hline 12 \\ 22 \\ 32 \\ 42 \\ \hline 13 \\ 23 \\ 33 \end{array}\right] = a_{43} \begin{bmatrix} u \\ v \end{bmatrix}$$

It is appropriate to set $a_{43} = 1$. Then the remaining a_{ij} are multiples of this choice.

This is what we will refer to as the "generic" form.

In order to solve for the a_{ij}, we need to furnish enough information so that the generic matrix on the left, consisting of two rows and eleven columns, becomes an 11 x 11 matrix, and then presumably can be inverted.

If six points are known in the $[x\ y\ z\ 1]$ space, and their images are measured in the $[u\ v]$ space, (the photograph) then twelve matric rows will be generated. This is one more row than we need. We have the option of omitting one row, or applying the method of least squares.

In either case, we now invert the matrix and find the a_{ij}.

Similarly, a second photograph is generated by

$$[x\ y\ z\ 1] \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} = [ks\ \ kt\ \ k]$$

and from this we can determine the b_{ij}.

Again we observe that typically

$$[x\ y\ z\ 1] \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix} u = hu,$$

which we now rewrite as

$$[x\ y\ z\ 1] \begin{bmatrix} a_{13}u \\ a_{23}u \\ a_{33}u \end{bmatrix} + a_{43}u = hu.$$

But also

$$hu = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + a_{41}$$

Now

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{13}u \\ a_{23}u \\ a_{33}u \end{bmatrix} + a_{43}u - \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} - a_{41} = 0,$$

or

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{13}u & - & a_{11} \\ a_{23}u & & a_{21} \\ a_{33}u & & a_{31} \end{bmatrix} = a_{41} - a_{43}u.$$

In order to solve for the $\begin{bmatrix} x & y & z \end{bmatrix}$ coordinates of any point, given the photograph measurements $\begin{bmatrix} u & v \end{bmatrix}$ and $\begin{bmatrix} s & t \end{bmatrix}$ we form the matrix equation

$$\begin{bmatrix} x & y & z \end{bmatrix} \left[\begin{array}{c|c|c|c} a_{13}u - a_{11} & a_{13}v - a_{12} & b_{13}s - b_{11} & b_{13}t - b_{12} \\ a_{23}u \quad a_{21} & a_{23}v \quad a_{22} & b_{23}s \quad b_{21} & b_{23}t \quad b_{22} \\ a_{33}u \quad a_{31} & a_{33}v \quad a_{32} & b_{33}s \quad b_{31} & b_{33}t \quad b_{32} \end{array}\right]$$

$$= \left[\begin{array}{c|c|c|c} a_{41} - a_{43}u & a_{42} - a_{43}v & b_{41} - b_{43}s & b_{42} - b_{43}t \end{array}\right].$$

The left-hand post-multiplying matrix is known, since for any point to be measured, $\begin{bmatrix} u & v \end{bmatrix}$ and $\begin{bmatrix} s & t \end{bmatrix}$ are known from the photographs.

It is, however, a 3×4 matrix. This simply means that it contains more information than is needed. We have the option of discarding one of the columns of this matrix, or of using again the least-squares procedure.

If we discard one of the columns, it is as though we considered one-and-a-half photographs sufficient. But of course this is indeed the case when we have two orthographic views of an object. One or the other of the two projections contains redundant information.

Burton's algebra was designed for a somewhat different case. He, essentially, had three "cameras", each recording a single coordinate instead of the two recorded by an ordinary camera. However, the algebraic tricks he used are the same as used here.

The reader might entertain himself by reconstructing Burton's procedure, based on the ideas presented here.

IMPLEMENTATION.

Of the several points measured in the $[\,x\ y\ z\,]$ object space, one point may be taken as the origin of temporary coordinates, since in the "real" space it is merely a translate.

Similarly, in the $[\,u\ v\,]$ (and $[\,s\ t\,]$) picture coordinates, these image points can be taken as the origins of these systems.

When we introduce these vector quantities into the generic expression, there results, for $[\,u\ v\,]$,

$$\left[\begin{array}{cccc|cccc|cccc} 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \end{array}\right] \left[\begin{array}{c} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \\ \hline a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \\ \hline a_{13} \\ a_{23} \\ a_{33} \end{array}\right] = a_{43} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which tells us immediately that $a_{41} = 0$ and likewise $a_{42} = 0$. Similarly, of course, $b_{41} = 0$ and $b_{42} = 0$.

Under these quite reasonable conditions, the general (restricted) form of the generic equation becomes

$$\left[\begin{array}{ccc|ccc|ccc} x & y & z & 0 & 0 & 0 & -ux & -uy & -uz \\ 0 & 0 & 0 & x & y & z & -vx & -vy & -vz \end{array}\right] \left[\begin{array}{c} a_{11} \\ a_{21} \\ a_{31} \\ \hline a_{12} \\ a_{22} \\ a_{32} \\ \hline a_{13} \\ a_{23} \\ a_{33} \end{array}\right] = a_{43} \left[\begin{array}{c} u \\ v \end{array}\right]$$

There are now only nine columns, and we need nine rows to make a square matrix, presumably invertible. This suggests that we need to measure five points in $\left[\, x \;\; y \;\; z \,\right]$ referred to the temporary origin, and the corresponding images in $\left[\, u \;\; v \,\right]$ referred to their origin. Similarly for $\left[\, s \;\; t \,\right]$.

Again, there will result more equations than we need, and we can either discard a row of the matrix, or use least-squares.

We choose $a_{43} = 1$, which yields a computable matrix. The result is:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} hu & hv & h \end{bmatrix}$$

and of course similarly for $\begin{bmatrix} ks & kt & k \end{bmatrix}$.

These a_{ij} (and the similar b_{ij}) are the transformation matrices that relate the object space to the image space.

PRODUCTS OF POLYNOMIALS.

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

For illustrative purposes, let it be required to form the product of a quadratic form

$$a_o + a_1 x + a_2 x^2$$

and a cubic form

$$b_o + b_1 x + b_2 x^2 + b_3 x^3 .$$

We know that the result will be a quintic form,

$$c_o + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5$$

and we wish to find the c_i.

We write:

$$\begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & b_3 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & b_3 \end{bmatrix}$$

and this matric product is the vector

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix}$$

In this, we construct the post-multiplying matrix in "row-eschelon" form, so that it has as many rows as the pre-multiplying vector (or matrix) has columns.

As a trivial example, construct the matrix of coefficients for $(1+x)^3$.

We write

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix}$$

a result we of course anticipated.

CONSTRAINED LEAST-SQUARES

by

S. A. COONS

Computer and Automation Institute

Hungarian Academy of Sciences

The theory of "least-squares", the approximation of a set of data points by a function intrinsically incapable of fitting the data exactly, is well-known.

Nevertheless, we shall develop this theory again, because we wish to show how to satisfy the least-squares criterion <u>and</u> the exact fit of the function to certain of the data points.

We shall, later on, consider a particular version of the problem, a special case, which will reveal and illuminate the details of the mechanism.

But first, let there be a set of data points, x_i, and a set of approximants to the data points, x_i^*. We say that the discrepancy between the approximant and the data points is the so-called residual,

$$r_i = x_i^* - x_i \quad .$$

Now, for a choice of a basis order 4,

$$x_i^* = \begin{bmatrix}\Phi_1(u_i) & \Phi_2(u_i) & \Phi_3(u_i) & \Phi_4(u_i)\end{bmatrix} \qquad \begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4\end{bmatrix}$$

Usually, but not necessarily, the basis is a set of monomial powers of the independent variable u_i, and we might have

$$\begin{bmatrix}\Phi_1(u_i) & \Phi_2(u_i) & \Phi_3(u_i) & \Phi_4(u_i)\end{bmatrix} = \begin{bmatrix}u_i^3 & u_i^2 & u_i^1 & u_i^0\end{bmatrix}$$

The α_j are coefficients of these elements, and it is these α_j that we wish to determine, so as to satisfy the least-squares criterion.

Let r_i be the column vector $\begin{bmatrix}r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot\end{bmatrix}$

We choose to abbreviate this, omitting subscripts, so that $r^T r$ = sum of the squares of the residuals. This is just

$$\begin{bmatrix}r_1 & r_2 & r_3 & \cdots\end{bmatrix} \qquad \begin{bmatrix}r_1 \\ r_2 \\ r_3 \\ \vdots\end{bmatrix}$$

We wish to minimize $r^T r$. In order to do this, we write

$$(r^T r)_\alpha = 0$$

to mean that we take partial derivatives with respect to each of the α_j and form a matrix equation which vanishes.

We observe that, accordingly,

$$(r^T r)_\alpha = r_\alpha^T r + r^T r_\alpha = 0$$

and, since the terms of this equality are mutual transposes, we can choose

$$r_\alpha^T r = 0 \quad \text{(The choice is immaterial.)}$$

Now $\quad r = \left[x^* - x \right]$, $\quad$ and r is a column vector.

$\quad r_\alpha = x^*_\alpha$ $\quad$ since x, the data point vector is constant.

But $\quad x^* = \Phi_\alpha$, $\quad$ a compact way of describing the x^* column vector.

We now have

$$r_\alpha^T = x^{*T}_\alpha = (\Phi\alpha_\alpha)^T \; .$$

(Later we will prefer $\alpha_\alpha^T \Phi^T$).

But $r_\alpha^T r$ is now

$$\begin{aligned} (\Phi\alpha_\alpha)^T r &= (\Phi\alpha_\alpha)^T (x^* - x) \\ &= (\Phi\alpha_\alpha)^T (\Phi\alpha - x) = 0, \end{aligned}$$

and this yields

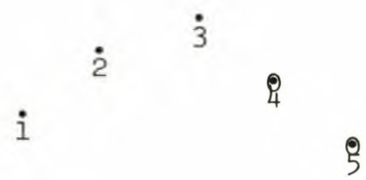
$$(\Phi\alpha_\alpha)^T \Phi\alpha - (\Phi\alpha_\alpha)^T x = 0$$

or $$(\Phi\alpha_\alpha)^T\Phi_\alpha = (\Phi\alpha_\alpha)^T x \ .$$

If the α (column vector) consists of strictly <u>linearly independent</u> α_j, then the <u>matrix</u> $\alpha_\alpha = I$, the identity, and we have the well-known least-squares equation,

$$\Phi^T\Phi\alpha = \Phi^T x.$$

But we wish to have "hard" constraints on the approximant. To show what happens in such a case, let us consider the special situation with five points,



and the stipulation that the function be of <u>order 4</u>, and that it must be constrained to fit points 4 and 5 exactly.

Evidently we can say that there is a <u>two-parameter pencil</u> of functions that pass exactly through points 4 and 5, and have two adjustable parameters that yield least-square-sums of the residuals at points 1, 2 and 3.

First, we need to know how the "hard" constraints of points 4 and 5 cause the α_j to be no longer linear independent.

For these "hard" constraints,

$$\begin{bmatrix} \Phi_1(u_4) & \Phi_2(u_4) & \Phi_3(u_4) & \Phi_4(u_4) \\ \Phi_1(u_5) & \Phi_2(u_5) & \Phi_3(u_5) & \Phi_4(u_5) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}$$

Writing typically, $\Phi_1(u_4) = \Phi_1{}^4$, (for notational simplicity), we can form a square invertible matrix, and then

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \Phi_1{}^4 & \Phi_2{}^4 & \Phi_3{}^4 & \Phi_4{}^4 \\ \Phi_1{}^5 & \Phi_2{}^5 & \Phi_3{}^5 & \Phi_4{}^5 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \left[\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \hline \alpha_4 \\ \alpha_5 \end{array}\right]$$

We can invert this square matrix, appropriately multiply, and collect constants, to obtain

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix} .$$

In our present case, let us write out in detail what we have found:

We have $$\alpha_\alpha^T \Phi^T \Phi \alpha = \alpha_\alpha^T \Phi^T x \quad .$$

But

$$\alpha = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix}$$ from the x_4 x_5 point constraints.

So

$$\alpha_\alpha = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix},$$

when we take columnar partial derivatives with respect to α_1 and α_2.

The left hand product $\alpha_\alpha^T \Phi^T \Phi \alpha$ is, for this special case,

$$\begin{bmatrix} 1 & 0 & a & d \\ 0 & 1 & b & e \end{bmatrix} \begin{bmatrix} \Phi_1 1 & \Phi_1 2 & \Phi_1 3 \\ \Phi_2 1 & \Phi_2 2 & \Phi_2 3 \\ \Phi_3 1 & \Phi_3 2 & \Phi_3 3 \\ \Phi_4 1 & \Phi_4 2 & \Phi_4 3 \end{bmatrix} \begin{bmatrix} \Phi_1 1 & \Phi_2 1 & \Phi_3 1 & \Phi_4 1 \\ \Phi_1 2 & \Phi_2 2 & \Phi_3 2 & \Phi_4 2 \\ \Phi_1 3 & \Phi_2 3 & \Phi_3 3 & \Phi_4 3 \\ \Phi_1 4 & \Phi_2 4 & \Phi_3 4 & \Phi_4 4 \end{bmatrix}$$

$$* \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix} \right).$$

Now we notice that the pre-multiplier of

$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$ is just

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ d & e \end{bmatrix}$$ which we have found to be α_α.

Accordingly, we can write for the left-hand product,

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha + \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \cdot \quad \alpha_\alpha^T \Phi^T \Phi \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix}$$

Now, fortunately $\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha$ is not only square, but it is non-singular, and therefore presumably invertible. The complete matrix equation is

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \alpha_\alpha^T \Phi^T \Phi \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix}$$

$$= \alpha_\alpha^T \Phi^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

We rewrite this as

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \alpha_\alpha^T \Phi^T \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \Phi \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix} \right)$$

Of course this matrix equation can be solved for

$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$ since $\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha$ is an invertible matrix.

And when we have found $\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$, we can write $\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}$, the desired set of coefficients.

These coefficients satisfy the hard constraints of x_4 and x_5, and give the best least-squares fit of x_1, x_2 and x_3, as required.

- . -

Now we are in a position to review the general least-squares equation. It is

$$\alpha_\alpha^T \Phi^T \Phi \alpha = \alpha_\alpha^T \Phi^T X.$$

When the α_j are strictly linearly independent, then $\alpha_\alpha = I$, an identity matrix, (as has been pointed out before) and the equation reduces to

$$\Phi^T \Phi \alpha = \Phi^T x,$$

the familiar un-constrained least-squares case.

But when the α_j are linearly constrained, we must look at the more general form.

It is

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \alpha_\alpha^T \Phi^T \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \Phi \begin{bmatrix} 0 \\ 0 \\ c \\ f \end{bmatrix} \right),$$

for our present case, or even more generally,

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha \alpha^* = \alpha_\alpha^T \Phi^T \ (X - \Phi K)$$

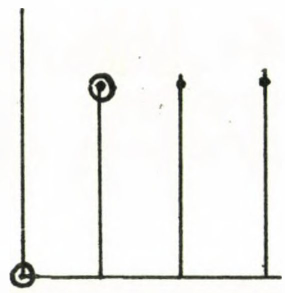
where α^* is intended to mean the vector of strictly independent α_j parameters, and K is intended to mean the columnar vector of constraints.

Φ is intended to mean the matrix of values for each of the approximant functions, not including those values that are involved in the constraint equations.

The product $\Phi\alpha_\alpha$ and its transpose $\alpha_\alpha^T \Phi^T$ appears in the equation, and can be evaluated once and for all.

CONSTRAINED LEAST-SQUARES.

TWO TESTS



Four points, and a quadratic function to fit.

The two hard points are x_0 and x_1, with describing vector

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

Assume the function is $x^* = \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$.

Then $$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

Augment the matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ 1 \end{bmatrix} .$$

Invert and solve:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \alpha_1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\alpha_\alpha = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}.$$

The soft point conditions at u = 2 and u = 3 are:

$$\begin{bmatrix} 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} x_2^* \\ x_3^* \end{bmatrix}$$

The equation is, in general:

$$\boxed{(\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha) = \alpha^T \Phi^T (x - \Phi k)}$$

where $\Phi = \begin{bmatrix} 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix}$, and $k = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$,

and $\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, the data points (soft).

$$\alpha_\alpha^T \Phi^T = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 9 \\ 2 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 6 \end{bmatrix}$$

$$\alpha_\alpha^T \Phi^T \Phi \alpha_\alpha = \begin{bmatrix} 2 & 6 \end{bmatrix} \begin{bmatrix} 2 \\ 6 \end{bmatrix} = 40$$

$$x - \Phi k = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\alpha_\alpha^T \Phi^T (x - \Phi k) = \begin{bmatrix} 2 & 6 \end{bmatrix} \begin{bmatrix} -1 \\ \hline -2 \end{bmatrix} = -14.$$

$$40\ \alpha_1 = -14$$

$$\alpha_1 = -.35$$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} -.35 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -.35 \\ 1.35 \\ 0 \end{bmatrix}.$$

Now to test the results:

$$\left[\begin{array}{ccc} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \hline 4 & 2 & 1 \\ 9 & 3 & 1 \end{array}\right] \begin{bmatrix} -.35 \\ 1.35 \\ 0 \end{bmatrix} = \left[\begin{array}{c} 0 \\ 1 \\ \hline 1.3 \\ .9 \end{array}\right] = \left[\begin{array}{c} x_0 \\ x_1 \\ \hline x_2^* \\ x_3^* \end{array}\right] \begin{array}{l} \text{exact} \\ \hline \text{approximate.} \end{array}$$

The residuals are:

$$\begin{bmatrix} 0 \\ 1 \\ 1.3 \\ .9 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ .3 \\ -.1 \end{bmatrix}.$$

The sum of the squares of these residuals is

$$(.3)^2 + (-.1)^2 = .09 + .01 = .1,$$

presumably a minimum.

We can test further, by perturbing the α quantities slightly.

Suppose $\alpha_1 = -.349$

$\alpha_2 = 1.349$

a pair of quantities that still satisfy the hard point conditions.

Then $$\begin{bmatrix} 4 & 2 \\ 9 & 3 \end{bmatrix} \begin{bmatrix} -.349 \\ 1.349 \end{bmatrix} = \begin{bmatrix} 1.302 \\ .906 \end{bmatrix}$$

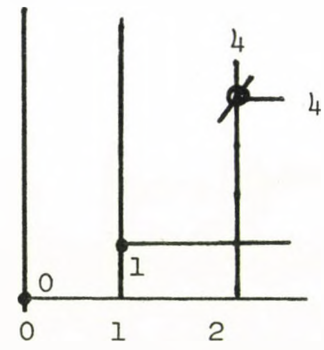
The residuals are $\begin{bmatrix} .302 \\ -.094 \end{bmatrix}$, and the sum of the squares is .10004. This is slightly larger than .1.

Again, perturb α_1 and α_2 upward:

$$\begin{bmatrix} 4 & 2 \\ 9 & 3 \end{bmatrix} \begin{bmatrix} -.351 \\ 1.351 \end{bmatrix} = \begin{bmatrix} 1.298 \\ .894 \end{bmatrix}$$

Residuals: $\begin{bmatrix} .298 \\ -.106 \end{bmatrix}$, and sum of squares is .10004, again slightly larger than .1, which can be taken to be the minimum.

We can choose hard constraints in other ways.



Here we will try to fit a quadratic so as to approximate points $\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and to pass exactly through point $x_2=4$, with a slope $x_2' = 4$.

The hard conditions are given by

$$\begin{bmatrix} u^2 & u & 1 \\ 2u & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

which is

$$\begin{bmatrix} 4 & 2 & 1 \\ 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

We may elect which of the α_i we wish to be the independent variable. In this case, choose α_2. The significance of this choice will appear shortly. Then, augmenting the matrix,

$$\begin{bmatrix} 0 & 1 & 0 \\ 4 & 2 & 1 \\ 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_2 \\ 4 \\ 4 \end{bmatrix} .$$

Inverting the matrix and solving,

$$\begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} = \begin{bmatrix}-1/4\\ 1\\ -1\end{bmatrix}\alpha_2 + \begin{bmatrix}1\\ 0\\ 0\end{bmatrix}$$

$$\alpha_\alpha = \begin{bmatrix}-1/4\\ 1\\ -1\end{bmatrix} \text{ and } k = \begin{bmatrix}1\\ 0\\ 0\end{bmatrix}$$

$$\Phi = \begin{bmatrix}0 & 0 & 1\\ 1 & 1 & 1\end{bmatrix} \text{ for the soft points,}$$

$$\Phi\alpha_\alpha = \begin{bmatrix}-1\\ -1/4\end{bmatrix}$$

$$\alpha_\alpha^T\Phi^T\Phi\alpha_\alpha = \begin{bmatrix}-1 & -1/4\end{bmatrix}\begin{bmatrix}-1\\ -1/4\end{bmatrix} = \frac{17}{16}\,.$$

$$x-\Phi k = \begin{bmatrix}0\\ 1\end{bmatrix} - \begin{bmatrix}0 & 0 & 1\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix} =$$

$$= \begin{bmatrix}0\\ 1\end{bmatrix} - \begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}$$

The general equation is now

$$\frac{17}{16}\;\alpha_2 = 0$$

$$\text{or } \alpha_2 = 0.$$

So $\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ The approximant function is just

$$x^* = u^2 .$$

And all residuals are exactly zero.

This case was purposely concocted so as to test the constrained least squares mechanism, to see whether it gave correct results when the proper solution is known.

The choice of α_2 as the independent variable was also a special trick, to see whether the mechanism worked for a "presumably improper" choice. The reader can verify that the choice of α_1 as the independent variable also gives the same result, and we might consider this choice a more "proper" one, since α_1 turns out to be 1.

Evidently the mechanism cannot be outwitted; it is apparently fail-safe and fool-proof.