

responsables de sección

Vida de la Sociedad

Mariano Gasca
Universidad de Zaragoza
gasca@posta.unizar.es
fax: (976) 35 62 44

Congresos y seminarios

Juan Luis Vázquez
Universidad Autónoma de Madrid
jvazquez@ccuam3.sdi.uam.es
fax: (91) 397 48 89

Novedades bibliográficas

Eduardo Casas
Universidad de Cantabria
sema@masun1.unican.es
fax: (942) 20 17 03

Actividad académica

Tomás Chacón
Universidad de Sevilla
chacon@cica.es
fax: (95) 455 28 98

Relaciones con la industria

Juan Manuel Viaño
Universidad de Santiago de Compostela
viano@zmat.usc.es
fax: (981) 59 70 54

LA PORTADA

La foto de portada muestra un aspecto de la Escuela de Estudios Universitarios de Vic, marco (junto con el Edificio de la Fábrica del Sucre) del último Congreso de Ecuaciones Diferenciales y Aplicaciones (CEDYA) / Congreso de Matemática Aplicada.

sumario

Editorial	2
Temas:	
<i>Automatic modeling of complex surface shapes</i>	
<i>using Bézier methods</i> , por Ramón F. Sárraga	4
<i>Former nos étudiants à la recherche</i> , por Claude Brezinski	33
Reseñas	40
La sociedad	43
Universidad	50
Libros	56
Resúmenes de tesis	60
Congresos y seminarios	68

edición

Editor jefe

FRANCISCO LISBONA CORTÉS
Departamento de Matemática Aplicada
Universidad de Zaragoza

Editores

GLORIA AGUILAR VILLA
Dpt. Matemática Aplicada
Centro Politécnico Superior
Universidad de Zaragoza

FCO.JAVIER SAYAS GONZÁLEZ
Dpt. Matemática Aplicada
Edificio de Matemáticas
Universidad de Zaragoza

Imprime: COMETA, S.A., carretera de Castellón, km. 3,400. 50013 Zaragoza
Dirección editorial: Departamento de Matemática Aplicada, Edificio de Matemáticas,
primera planta, Campus Universitario, 50009 Zaragoza.
e-mail: sema@posta.unizar.es Fax: (976) 76 11 25

En la Asamblea anual de SEMA, celebrada en Vic el pasado setiembre, coincidiendo con la celebración del XIV CEDYA-IV Congreso de Matemática Aplicada, fui elegido Presidente de la Sociedad para el curso 1995-96. Aunque ya allí agradecí a los presentes la confianza y el honor que se me otorgaban, quiero transmitir a todos, por estas líneas, el mismo agradecimiento y a la vez las mismas reflexiones respecto al funcionamiento y futuro de SEMA. Comparando con sociedades análogas de países de nuestro entorno, la nuestra es aún muy pequeña. Bien es cierto que se llevan todavía muy pocos años de funcionamiento, pero un ligero vistazo a los listados de los Departamentos de Matemática Aplicada de las Universidades españolas nos dice que el número de socios es incomparablemente menor que el de miembros de aquéllos.

Dos primeras razones se me ocurren rápidamente, sin pretender que sean las únicas: la primera es que tenemos mala tradición respecto al funcionamiento de sociedades matemáticas españolas. La segunda es que los que actualmente pertenecemos al área de Matemática Aplicada somos la unión de muchísimos grupos tan diversos y con tantas conexiones con otras áreas matemáticas que difícilmente podemos encontrar algo característico del área.

Sin embargo, estas razones, vistas a través de los pocos años transcurridos desde el principio de la sociedad, me hacen ser optimista respecto a su futuro. Respecto a la primera, porque hemos conseguido un cierto dinamismo tanto en los Consejos Ejecutivos como en la comunicación con los socios a través del Boletín. Esta sociedad no está vinculada especialmente a ninguna persona ni pequeño grupo, sino que es de todos, y más lo será cuanto más lo sintamos así y participemos. Y esto es especialmente cierto en lo que respecta al Boletín. El actual equipo redactor ha hecho un gran esfuerzo para mejorar su calidad en cada número y así lo habéis reconocido muchos de vosotros. Pero esa labor no puede ser realizada mucho tiempo por las mismas personas a no ser que encuentren gran colaboración en todos nosotros. No puede ser que los mismos que realizan las labores editoriales tengan además que solicitar las colaboraciones, pedirnos noticias de congresos, pedirnos resúmenes de tesis, etc. etc. Mientras cada uno no nos sintamos redactor y no les enviemos todo este tipo de noticias, trabajos de divulgación de nuestras líneas de investigación y de las de nuestros visitantes, etc., para que ellos seleccionen, en su caso, el Boletín no será todo lo

útil que debería.

Respecto a la dispersión de grupos dentro de la Matemática Aplicada, creo que ahí precisamente puede estar nuestra riqueza y que también será labor nuestra saber captar a personas que hasta este momento no se han sentido vinculadas. Entre otras cosas, para que los congresos CEDYA sean también de Matemática Aplicada, como oficialmente los llamamos, deben abrirse a temas que no sean exclusivamente ecuaciones diferenciales, sus aplicaciones y su resolución numérica.

Con la reglamentaria renovación anual de un tercio del Consejo Ejecutivo han dejado de pertenecer a éste Alfredo Bermúdez de Castro, Ildefonso Díaz y Jesús Sanz Serna. Tanto por su labor como miembros del Consejo como por haber formado parte del pequeño grupo que puso en marcha la Sociedad hace pocos años es de justicia agradecerles todo el esfuerzo dedicado. Para sustituirles fueron elegidos en Vic Luis Abia, Alfonso Casal y Juan Manuel Viaño.

El pasado día 1 de diciembre tuvo lugar en Madrid la primera sesión del nuevo Consejo, en el que se procedió a la renovación de los cargos de Vicepresidente y Secretario, en las personas de Gerard Gómez y Alfonso Casal respectivamente.

Otros acuerdos de la misma reunión fueron:

- Designar a la Universidad de Vigo como sede del próximo CEDYA-Congreso de Matemática Aplicada.

- Reorganizar las secciones del Boletín en la forma en que ya aparecen en el presente número, con el consiguiente cambio de responsables.

- Buscar un corresponsal del Boletín en cada Universidad que envíe las noticias científicas locales a las correspondientes secciones.

- Iniciar el estudio para proponer una reforma de los estatutos en lo que respecta a elección de los futuros Presidentes y a la extensión de su duración a dos años.

- Hubo una generosa oferta de Gerard Gómez para empezar a hacer funcionar un servicio de Noticias a través del correo electrónico. Como no todos los socios comunicasteis en su día la dirección para dicho correo, todos los interesados podéis contactar con él a través de gomez@cerber.mat.ub.es

MARIANO GASCA
PRESIDENTE DE SEMA

T-corners constitute a special situation during GC^1 smoothing of patches, because one must modify again the “dependent” edge (see above) that enters a T-corner. By contrast, non-dependent edges remain untouched after coplanarity is enforced. At a GC^1 T-corner, the dependent edge must become tangent to the patch that lies across the corner from the dependent edge. The acyclic graph used for building the T-corners must be followed in order to apply the modifications in a consistent fashion.

5 Critique of the Surface Construction Method

As explained in previous sections, Surfplan’s approach to surface construction is strictly local. The approach is based on small corrections applied independently in small regions of a curve grid (e.g., at a corner point and its incident edges). This approach is particularly effective in preventing holes and correcting other local errors. However, because there is no coordination between local corrections, the method cannot provide global features for the constructed surfaces. For example, the method cannot eliminate ripples that extend across several patches.

Nevertheless, Surfplan’s data-structure framework allows a user of the software to execute a large number of local modifications in a single user operation. One such application is “marrying” of surfaces, i.e., joining two existing Surfplan surface models into one new model, even though the original surfaces do not exactly match along their common boundary edges. Surfplan marries surfaces by performing many small corrections to the existing models at the locations where they fail to agree (assuming the differences are not too large). Execution of this process is automatic, as the software simply follows in an orderly fashion the paths available among the data entities created during surface construction. Although such data structure traversal could also form the basis for global surface modifications, Surfplan was never enhanced in this way.

Surfplan was implemented at General Motors using the Lisp programming language [Steele 84]. This language, which was invented around 1960 mainly for artificial intelligence applications, provided an excellent software platform for manipulating complex data structures, both in the geometric model described in this paper, and in the user interface menus, which are discussed briefly in [Oetjens et al. 90]. In particular, Lisp’s garbage collection mechanism freed Surfplan developers from the need to write code to manage space within the computer.

On the other hand, Lisp code executes numerical algorithms much more

**AUTOMATIC MODELING OF
COMPLEX SURFACE SHAPES
USING BÉZIER METHODS**

RAMÓN F. SÁRRAGA
ANALYTIC PROCESS DEPARTMENT
GENERAL MOTORS RESEARCH AND DEVELOPMENT CENTER
WARREN, MICHIGAN 28090-9055

1 Introduction

The field of computer aided geometric design (CAGD) originated in the 1950's and 1960's from the computer-based application of mathematics, mostly numerical analysis, to problems of geometric design in industry. Some of the earliest and most significant pioneers worked in the automotive industry. Persons like Paul de Casteljaou at Citroën, Pierre Bézier at Renault, and William Gordon at General Motors have left a legacy of concepts and methods that continue to be applied regularly in computer design systems. (See [Farin 93, Hoschek et al. 93] for comprehensive historical bibliographies.)

As computer performance and versatility have continued to improve dramatically over the last four decades, industrial research units have been motivated to develop a fair number of software systems that apply CAGD methods to new design problems, or to old problems in more efficient ways. The trend has been to involve more areas of mathematics than just numerical analysis. Similarly, computer software organization has become more sophisticated and thus more complicated and difficult to maintain. The driving motivation for this software effort has been to reduce design time by relegating to the computer an ever increasing number of the tasks that humans used to perform.

This paper describes the data structure organization of a small research software system, called *Surfplan*, which was developed at the General Motors Research and Development Center starting around 1985. The goal of the system was to automate as much as possible the design of certain tool surfaces that are critical to the stamping of sheet metal. These surfaces are characterized by

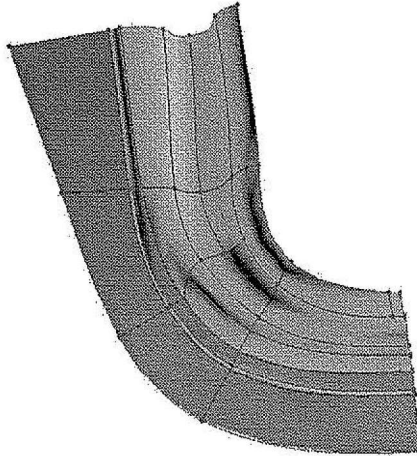


Fig. 1. Bézier-patch model of a region on the face of a die.

large changes in shape within relatively small regions, in contrast with car body surfaces, whose shape varies gently and gradually over much larger regions.

Figure 1 illustrates a tool surface of the type considered in this paper. The standard approach to modeling such tool surfaces has been (a) to design first a grid of curves, which captures the main features of the abrupt changes in shape, and then (b) to “cover” the curve grid with small surfaces, called patches, that interpolate or approximate all curves in the grid without leaving detectable holes. The engineering application does not require that the patches be joined with exact smoothness; therefore, the shape requirements are less demanding than those of car body surfaces.

However, the large number of patches needed to cover a typical grid makes the process very time consuming, if a designer must model each patch individually and must check every patch border for holes and other geometric anomalies. Moreover, the mathematical form of the patches must be compatible with that of production software systems used in manufacturing. In practice, this requirement means that the patch definitions have to be written into a file in a recognized format, such as [IGES 86] in North America.

The Surfplan system achieved a major reduction of design time for the tool surfaces in question by (a) using Bézier numerical methods [Bézier 86] and (b) supporting these methods with a novel data structure organization based on very elementary principles of algebraic topology (like the data structures used in the field of solid modeling). A report on Surfplan’s user interface and design

applications has been made public in [Oetjens et al. 90].

The main focus of this paper is to explain how Surfplan's data structures were designed in concert with Bézier numerical methods in order to support fairly automatic construction of surfaces. A didactic style at an introductory level is employed throughout the paper. Mathematicians whose computer experience is limited to straightforward calculations may find the data structure design to be instructive, as it illustrates an important way in which mathematics can be made useful to industry.

Section 2 of this paper introduces the essential components of the data structures that define a surface model in Surfplan. Section 3 gives an overview of how standard Bézier methods influence data structure design, while Section 4 summarizes the stages of surface construction that Surfplan automatically follows, unless an ambiguity is detected, or the user interrupts the process to make changes manually. Section 5 summarizes what Surfplan accomplished and what was left undone. Finally, the appendix outlines the major data structures of Surfplan's surface model.

2 Ingredients of the Surface Model

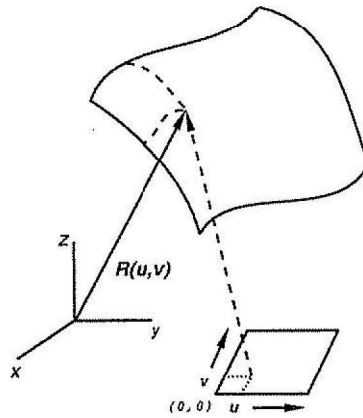


Fig 2. Parametric surface with a rectangular domain.

The basic ingredient of a surface model is a function, or set of functions, that specifies the points in R^3 where a modeled surface is located. The most common method [Faux et al. 79, Bartels et al. 88, Farin 93, Hoschek et al. 93] for specifying these equations is *parametric*, which means that each coordinate is expressed as an explicit function of two variables called *parameters*. The ar-

rament is illustrated in Figure 2 using a rectangular parameter domain. To every point (u, v) in the domain, there corresponds a point (x, y, z) in R^3 . As discussed in the next section, this correspondence is usually expressed either by a single polynomial (the Bézier case) or by piecewise polynomials (the B-spline case). (For an alternative method of surface definition, see [Cavendish & Marin 88].)

No matter what type of function is used for (x, y, z) in a parametric surface, one can show that many complicated physical surface shapes cannot be represented by a set of piecewise polynomials defined over a single rectangle. (See, e.g., [Veenman 82].) In fact, a well-known theorem of topology states that a mapping from the plane to any closed surface (e.g., a sphere) must have at least one singular point. In addition, one finds in practice that good surface fits become progressively more difficult to obtain as the number of physical surface features increases.

Therefore, realistic surface modeling systems are forced to deal with sets of parametric definitions, where each definition applies only to a region of a surface, and the entire surface is considered to be the totality of the regional definitions. In the subsequent discussion, a parametric definition of a surface region will be called a *patch*.

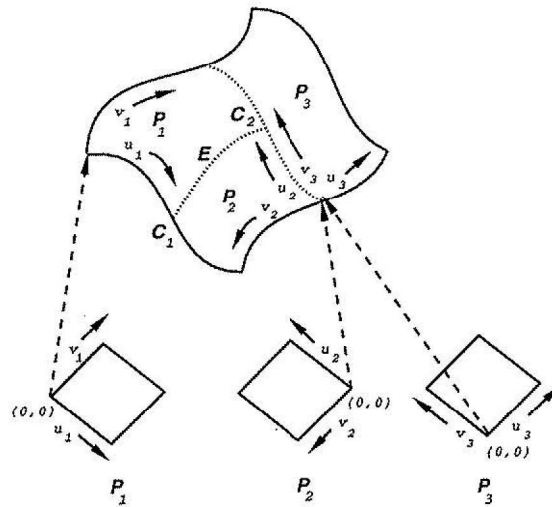


Fig 3. A surface with multiple patches.

Figure 3 illustrates a group of patches. Note that each patch has its own individual parametric definition, which *a priori* is independent of the parametric definition of any other patch. This fact raises the following issue: if the entire

surface is to have no holes, then adjacent patches must define the same curve along their common boundary (e.g., curve E in Figure 3). The requirement of no holes is generally called GC^0 smoothness in the field of CAGD for the reason explained below.

GC^0 smoothness would be an extremely rigid constraint in practice except for the fact that a curve in R^3 can have more than one equivalent parametric definition. In other words, the same set of (x, y, z) points on a curve can be represented by different functions of a parameter. In addition, as explained later, one can partition a curve into segments, each having its own independent parametric definition. Because of this multiplicity of representations, the term " C^0 smoothness" is generally restricted in CAGD to the case where adjacent curves or surfaces share the same parametric definition in addition to being continuous.

Clearly, a software system that handles surfaces with multiple patches must represent in its data base the mathematical features mentioned above. Because of the resulting complexity, system architects find it convenient to invent a system-specific language that assigns mnemonic names to the various types of mathematical entities in the surfacing system's data structure. Surfplan's basic entities are *patches*, *edges*, and *corners*. Patches are independent parametric definitions; edges are segments of curves bordering patches, and corners are (x, y, z) points at which edges meet. For example, edge E of Figure 3 is bounded by corners C_1 and C_2 .

Although explicitly representing edges and corners in the modeling database may seem redundant, in fact the presence of such entities simplifies algorithms for answering queries such as: *Are patches P_1 and P_2 adjacent?* or: *List in counterclockwise order the patches that meet at corner C_1 .* For example, in terms of the data structure, patches P_1 and P_2 are adjacent if they share a common edge data structure. Moreover, patches P_1 and P_2 do not need to exist in the numerical sense (i.e., as specific parametric functions) for their adjacency to be recorded.

Patch properties such as adjacency along an edge or at a corner are clearly topological in nature. Absence of holes between patches is also a topological property. Note that, if patches P_1 and P_2 do not meet along a common curve, one can denote this fact by simply having two edges between corners C_1 and C_2 , one edge connected to P_1 , and the other to P_2 .

In addition to topological properties, one can record numerical information along edges and at corners, viz., properties that quantitatively affect the shape of

the surface. For example, if patches P_1 and P_2 meet along E in such a way that their unit normals coincide at every point of E , this fact (called GC^1 smoothness) can be recorded in the edge data structure. Again, although this record may appear redundant, it allows the software to modify parametric definitions of patches without losing track of geometric constraints.

In addition to patches, edges, and corners, a comprehensive surface model should record *features*. A feature is a geometric entity with design or manufacturing significance, e.g., a reflection highlight curve on an outer body panel, a flange curve along which sheet metal is folded, or a localized construction such as a pocket. By contrast, patches, edges, and corners are merely artifices imposed by limitations of parametric surface definitions. They certainly do not constitute features and often are not created in such a way that every feature is composed of a unique set of patches, edges, and/or corners. Surfplan was not developed to the point of allowing operations on features.

3 Modeling Fundamentals

This section presents an overview of the properties of parametric curve and surface definitions most commonly used in surface modeling. The major purpose of the discussion is to justify the peculiarities of the data structures introduced in the previous section, viz., the patches, edges, and corners. Rather than deriving the results from mathematical expressions, the discussion here simply states properties of parametric definitions and then proceeds to develop the data structuring consequences. Note that the discussion applies equally well to NURBS (non-uniform rational B-splines), although it concentrates on polynomial and piecewise polynomial functions. (For an exposition on NURBS, see [Farin 93, Hoschek et al. 93].)

The main message of this section, from the point of view of data structure design, is how software complexity increases because surface modeling systems have to store multiple alternative representations of the same geometric object.

3.1 Example: Segmented Curves and T-Corners

Consider, for example, the patches in Figure 4, where patches P_1 and P_2 meet patch P_3 with GC^0 continuity (i.e., with no holes) along two edges, E_1 and E_2 . Assume that the patches are all of the Bézier type; i.e., they are polynomial rather than piecewise polynomial. Then E_1 and E_2 must represent segments of the same polynomial curve, because a border of a Bézier patch is always a polynomial curve, and P_3 borders E_1 and E_2 on the same side.

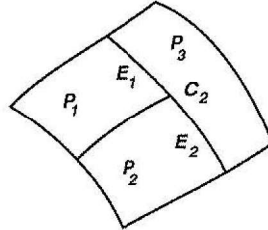


Fig 4. T-corner on patch boundaries.

Surfplan stores the fact that a sequence of edges (E_1, \dots, E_n) lies on a common curve by means of a special entity called a *segmented curve*. A segmented curve carries both the parametric definition of the entire curve and an ordered list of the curve's edges. In addition, the corners between edges on the same curve (C_2 in Figure 4) are called *T-corners* and carry information to distinguish them from regular corners.

Note that the surface modeling system carries multiple parametric definitions of the curve segment E_1 : (a) the parametric definition stored in the segmented curve, (b) the parametric definition stored in the segment E_1 itself, (c) the curve definition inferred from the definition of patch P_1 , and (d) the curve definition from patch P_3 . Obviously, the modeling system must ensure that these definitions agree with each other.

The convention adopted in Surfplan's current applications is that the definition of the segmented curve has top precedence. (A later section will show how this definition is calculated.) Then the definitions in E_1 and E_2 are obtained by means of recursive subdivision [Boehm 80, 81]. Finally, the patch side of P_2 is obtained from the definition in E_1 , while the patch side of P_1 comes from the segmented curve.

Using this convention, it is straightforward to propagate changes made to a segmented curve, in such a way that GC^0 smoothness is not violated in the surface model.

3.2 Recursive Subdivision (Knot Insertion)

Although a comprehensive discussion of recursive subdivision is beyond the scope of this report (see [Boehm 80, 81]), the fundamental meaning of this technique and its impact on surface modeling can be explained in a simple way.

Consider the case of a polynomial space curve in R^3 , i.e., a curve described

by a polynomial

$$R(t) = A_0 + A_1t + A_2t^2 + \dots + A_nt^n,$$

where $R(t)$ and the coefficients A_i are points in R^3 . (This representation is called the *power representation* or *power basis*.) In computer aided geometric design one usually imposes a restriction on t , such as

$$0 \leq t \leq 1.$$

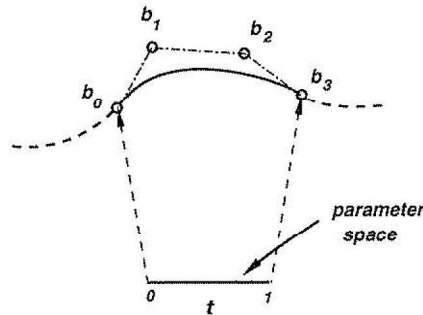


Fig 5. Parametrized segment on a curve ("wire").

However, the mathematical curve itself, for any t , ranges over space like a "wire", represented by dashes in Figure 5, and a segment of the "wire" is selected as the *image* of the straight-line parameter segment $0 \leq t \leq 1$. Note that the position and extent of this image segment on the "wire" is controlled by the A_i coefficients.

Recursive subdivision is merely a technique for changing the position and extent of an already specified image segment on the "wire". (The change in position is not usually discussed in the introductory literature.) Subdivision provides a set of mathematical equations relating the A_i coefficients of a new segment to those of an earlier segment. Thus, in the case of a segmented curve that gives rise to two edges E_1 and E_2 , subdivision provides a method for deriving the control points of the segments from the control points of the curve, thereby guaranteeing numerical agreement between the curve and its segments. (See Figure 6.)

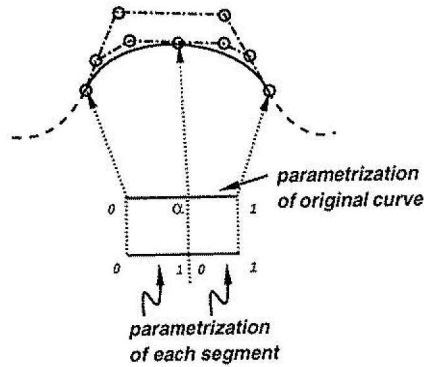


Fig 6. Subdividing a parametrized segment in two.

3.3 Bézier Curves

The well-known Bézier representation of a “wire” curve segment is merely a rearrangement of the polynomial powers shown earlier in the form

$$R(t) = b_0(1-t)^n + b_1 \left[\frac{n}{(n-1)} \right] t(1-t)^{n-1} + \dots + b_n t^n,$$

with $0 \leq t \leq 1$ and $R(t), b_i \in R^3$. This representation is called the *Bernstein basis*. Its value in CAGD stems from the geometric positions of the b_i coefficients relative to the image curve segment, as shown in Figure 5 for a cubic curve ($n = 3$). Note that b_0 and b_n lie at the end points of the curve segment, and that the vectors $b_1 - b_0$ and $b_n - b_{n-1}$ meet the segment tangentially at its end points.

Another property of Bézier curves that is used often in surface modeling is *degree elevation*. This is a method for representing a curve of degree n in the power basis using a Bernstein basis of higher degree. The two curves remain the same “wire” segment, because the coefficients of the higher powers cancel out when the Bernstein basis is transformed back to the power representation.

One useful application of degree elevation is to represent curves of lower degree as boundaries of patches of higher degree. This application is used often in Surfplan and provides another example of the need to maintain multiple representations of the same geometric object within a surface modeler.

3.4 B-Spline Curves

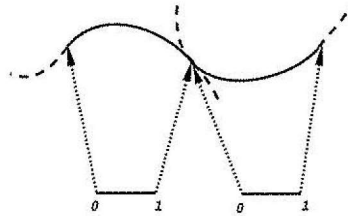


Fig 7. Hooking "wires" to form a piecewise polynomial curve.

By contrast with the single polynomial, a piecewise polynomial curve consists of several "wires" hooked together in such a way that they meet each other with some user-specified degree of smoothness, such as the tangentially smooth connections of Figure 7.

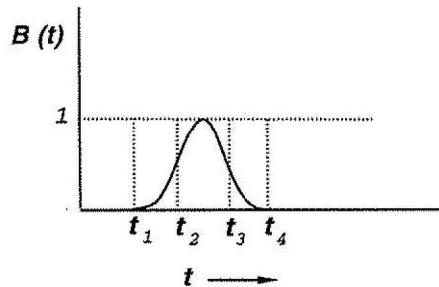


Fig 8. Graph of a single B-spline function of degree 2.

Except for special cases, design systems generally avoid the error-prone book-keeping needed to ensure that all polynomial pieces remain smoothly connected in a piecewise polynomial curve. A very effective solution to this problem is the use of B-splines. A single B-spline function is a bell-shaped, already pre-connected piecewise polynomial function (Figure 8). Notice that the function is zero everywhere except in a finite interval.

Each of the (x, y, z) space coordinates of a B-spline curve is represented as a linear combination of such B-spline functions operating on the same parameter t with the same types and positions of the breaks. In addition, the B-splines are required to overlap according to certain prescriptions, as illustrated schematically in Figure 9. The figure is only conceptual, since the overlap relationship is not simple to portray in a geometric fashion.

Recursive subdivision and degree elevation are also available for piecewise polynomial curves represented in terms of B-splines. Recursive subdivision is

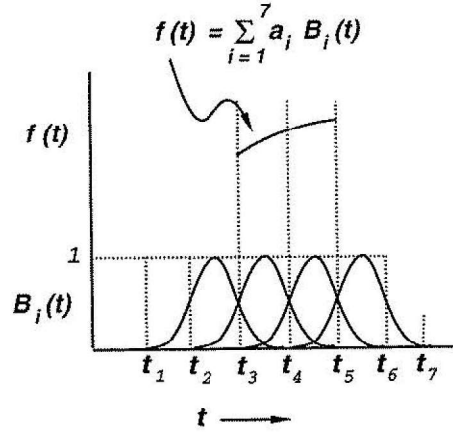


Fig 9. Overlapping B-splines of degree 2.

performed using a technique called *knot insertion*. The details of knot insertion are essentially irrelevant to this paper. (See [Farin 93])

3.5 Bézier and B-Spline Surfaces

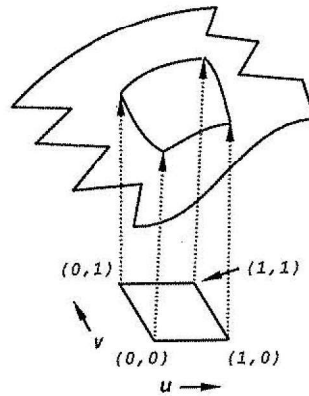
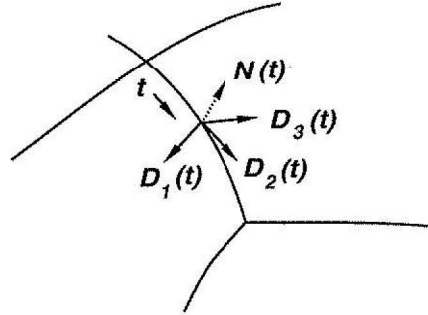


Fig 10. Parametrized region on a surface ("sheet").

The basic facts presented earlier for curves apply also to surfaces, *mutatis mutandis*. Instead of a "wire", one can visualize a sheet in space. A patch on the sheet is now an image of the unit square with parameters $0 \leq u \leq 1$ and $0 \leq v \leq 1$, as shown in Figure 10. Subdivision can be used to adjust the position and extent of a patch on the sheet, but each application of subdivision must

be performed in only one parametric variable at a time, i.e., u or v . In other words, a diagonal subdivision, for example, is not defined.



$$N = \frac{D_1 \times D_2}{|D_1 \times D_2|} = \frac{D_2 \times D_3}{|D_2 \times D_3|}$$

Fig 11. GC^1 smoothness between patches.

There also exist triangular patches, but no fully satisfactory B-spline theory is known for them. Moreover, even for rectangular patches, there exist restrictions on how the patches can be connected with GC^1 smoothness to each other, i.e., so that the unit surface normals of adjacent patches coincide on their common boundary, as illustrated in Figure 11.

4 Stages in Building Surfaces from Digitized Curves

The original application of Surfplan in General Motors has been to construct surfaces from digitized point data. The data points are arranged in sequences, where each sequence represents a conceptual curve. Typically, the points on a curve are obtained from pencil markings on a physical model, usually a plaster prototype. The markings delineate the boundaries of patches to be created. Surfplan does not receive any data for numerically approximating patches inside their boundaries.

From the description below, it will be clear that the process of marking curves on a physical model must be performed by a trained person who understands the limitations imposed by the use of cubic Bézier curves. However, only a qualitative geometric understanding is required, not knowledge of the underlying mathematics.

4.1 Initial Curve Fitting

After reading the digitized data that represents potential patch boundary curves, Surfplan performs a least-squares fit of each set of curve points using a parametric cubic curve in Bézier form, unless the number of curve points is less than four. In the latter case, the curve is generally interpolated as a quadratic (three points) or a straight line (two points). The word “generally” alludes to data fixing operations (e.g., throwing away points that are too close to each other), which do not concern this paper.

After creating a unique edge data structure for each fitted curve, Surfplan proceeds to group endpoints of edges into corner structures. Roughly speaking, endpoints that are within a user-supplied tolerance distance belong to the same corner. However, Surfplan checks for looping edges (when both endpoints of an edge are initially assigned by the software to the same corner) and, in this case, Surfplan redoes the computation with a smaller tolerance in order to avoid the loop. During the process, endpoints are sorted in space in order to avoid testing all pairs of endpoints.

Each created corner data structure is assigned a unique location in space equal to the average point of the constituent endpoints. Then all incident edge curves are modified so that their endpoints coincide with the corner location. For future reference, the original digitized points are stored in association with the corresponding edge data structure.

4.2 Resolving T-Corners

At this point in the process, if there are corners with just one edge attached, each such corner is checked to see if it lies close to the interior of some other edge, i.e., if it is a T-corner candidate. This check should find a unique minimum-distance edge for the corner in question, where the distance between a corner and an edge is defined as the distance between the corner and the closest point on the edge. If more than one edge point is at the closest distance from the corner, an error is issued: the user must handle the situation interactively. The minimum-distance edge is used to form a T-corner only if the minimum distance is within a specified tolerance. A simple application of Newton’s method is used to find the parameter value of the edge point closest to the corner.

Note that corners containing more than one attached edge are not tested for proximity to the interior of an existing edge. The T-corners described above are a unique case in Surfplan. In all other cases, an input curve is assumed

4.1 Initial Curve Fitting

After reading the digitized data that represents potential patch boundary curves, Surfplan performs a least-squares fit of each set of curve points using a parametric cubic curve in Bézier form, unless the number of curve points is less than four. In the latter case, the curve is generally interpolated as a quadratic (three points) or a straight line (two points). The word “generally” alludes to data fixing operations (e.g., throwing away points that are too close to each other), which do not concern this paper.

After creating a unique edge data structure for each fitted curve, Surfplan proceeds to group endpoints of edges into corner structures. Roughly speaking, endpoints that are within a user-supplied tolerance distance belong to the same corner. However, Surfplan checks for looping edges (when both endpoints of an edge are initially assigned by the software to the same corner) and, in this case, Surfplan redoes the computation with a smaller tolerance in order to avoid the loop. During the process, endpoints are sorted in space in order to avoid testing all pairs of endpoints.

Each created corner data structure is assigned a unique location in space equal to the average point of the constituent endpoints. Then all incident edge curves are modified so that their endpoints coincide with the corner location. For future reference, the original digitized points are stored in association with the corresponding edge data structure.

4.2 Resolving T-Corners

At this point in the process, if there are corners with just one edge attached, each such corner is checked to see if it lies close to the interior of some other edge, i.e., if it is a T-corner candidate. This check should find a unique minimum-distance edge for the corner in question, where the distance between a corner and an edge is defined as the distance between the corner and the closest point on the edge. If more than one edge point is at the closest distance from the corner, an error is issued: the user must handle the situation interactively. The minimum-distance edge is used to form a T-corner only if the minimum distance is within a specified tolerance. A simple application of Newton’s method is used to find the parameter value of the edge point closest to the corner.

Note that corners containing more than one attached edge are not tested for proximity to the interior of an existing edge. The T-corners described above are a unique case in Surfplan. In all other cases, an input curve is assumed

to be digitized only between nearest corners, even if the curve has a tangent continuation curve extending beyond a corner.

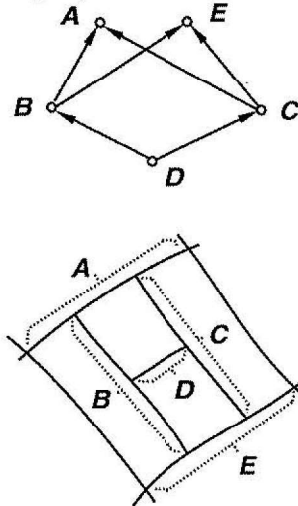


Fig 12. T-corner dependency graph and its edge geometry.

T-corners give rise to a special dependency relation among edges. For example, in Figure 12, the final shape of edge D cannot be determined before the shapes of edges B and C are known. These edges, in turn, depend on edges A and E . If the dependency relation is not circular, it defines a partial ordering of edges. The ordering can be represented by a directed acyclic graph [Deo 74], as shown in Figure 12. Each node of the graph represents an edge of the surface. A directed edge of the graph, " $B \rightarrow A$ ", denotes the relation " B depends on A ."

Surfplan issues an error if the edge dependency relation turns out to be circular. This situation is illustrated in Figure 13, where the dependency chain runs: $A \rightarrow D \rightarrow C \rightarrow B \rightarrow A$.

When a T-corner is created, the dependent edge is modified to ensure that it numerically touches the edge on which it depends. In fact, every time that an edge with T-corners is modified, the modifications are propagated down (i.e., opposite to the directions of the arrows) through the subgraph rooted at the current edge.

4.3 Grouping Edges into Patches

Automatic patch creation is basically just a matter of finding closed paths starting at a corner and traversing four edges to return to the initial corner. However,

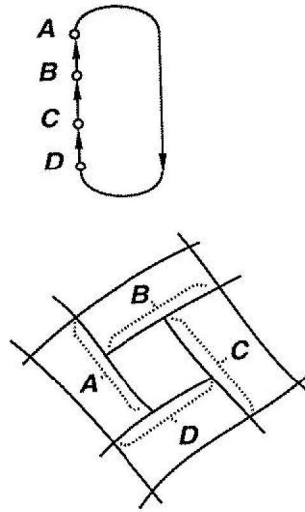


Fig 13. Cycle in a T-corner edge dependency.

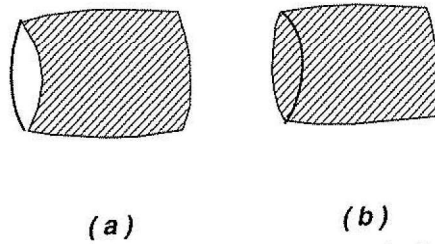


Fig 14. Possible patches are shaded.

the task is complicated by the presence of T-corners, which are counted in a different manner depending on how the path enters and exits from the T-corner. In theory, there also are ambiguous configurations of edges, such as the one shown in Figure 14, where patch topology cannot be inferred from curve topology, e.g., to distinguish between patch configurations (a) and (b). This example is detected as an "error" in the created surface (a hole in case (a) or a "loose" edge in case (b)), and the user is asked to correct the problem manually.

After the edges bordering each patch are identified, interior control points are estimated via "bilinear Coons blending" [Faux et al. 79]. At the end of this stage, all patches are tensor-product (rectangular) Bézier patches of degree $m \times n$, with $m, n \leq 3$. Although patches have been created independently of each other, adjacency is now recorded in the data structures.

4.4 Orienting the Surface

Orienting the surface does not involve any numerical computation or change of surface shape; rather, it is a reordering of data structures throughout the entire surface model in order to introduce a standard set of relationships among the constituents of the surface model.

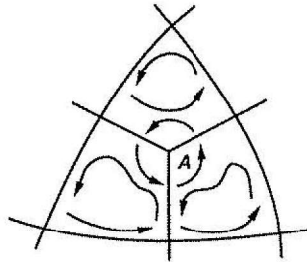


Fig 15. Surface orientation.

The process is illustrated in Figure 15 for patches and edges meeting at a common corner. In the case of patches, define a *standard circuit* around a patch as a traversal of the patch sides in the order 1 to 4, as explained in the appendix and illustrated by the arrows inside the patches in Figure 15. Surface orientation simply stipulates that these arrows must point in opposite directions along the common edge of any two adjacent patches, as in Figure 15. This prescription ensures that, if adjacent patches are joined in a GC^1 manner, patch normals calculated at the same space point on the common boundary will point in the same direction. (Surfplan does not require GC^1 joining of adjacent patches.)

Now consider corner *A* in Figure 15. Surfplan's surface orientation step ensures that the edges are ordered in the order in which one would encounter them as one moves around corner *A* following the circular arrow. If corner *A* has a unique surface normal, the circular arrow winds around it in the counterclockwise direction.

In the following discussion, any ordering of data structures motivated by surface orientation is called a *canonical order*.

Surface orientation is not merely a convention: it facilitates the implementation of algorithms that operate on sets of geometric data entities, e.g., on sets on patches. Historically, elementary principles of algebraic topology, such as orientation, were first applied to data structures in solid modeling systems. Their application to surface modeling avoids an important family of potential

inconsistencies that plagued early surface modeling approaches.

4.5 Ensuring Tangency of Edges at Corners

Ensuring tangency of edge curves is mostly a corrective measure that enforces tangency for curves that were marked as tangent on the physical model but lost their tangency because of digitization, least-squares fitting, matching endpoints at common corners, and T-corner fitting.

In general, edges whose tangents form an angle less than a user-supplied tolerance are made numerically tangent by averaging their tangent directions but ensuring they touch the corner.

In practice, users generally want to make tangent some edge pairs whose angle is greater than the general tolerance. Surfplan provides a user interface to display instances of edges with angles within a specified range above tolerance, so that the user can select a set of instances from the screen and have Surfplan correct these as a group.

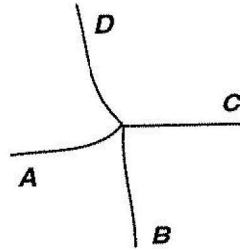


Fig 16. Case where nontangent edges exclude GC^1 smoothness.

Enforcing tangency of edges is a necessary prerequisite for GC^1 smoothness of patches in certain configurations of edges and patches. For example, in Figure 16 all four patches that meet at the corner cannot be made pairwise GC^1 along their common edges, unless alternating edges are mutually tangent, viz., A with C, and B with D. (For details, see [Bézier 86] and [Sarraga 87, 90]; more recently, Peters has shown that higher-derivative conditions imposed at the corner will achieve GC^1 continuity of the patches [Peters 91].)

4.6 Ensuring Coplanarity of Edges at Corners

GC^1 smoothness of patches also requires that certain edges have a common tangency plane at appropriate corners. For example, the three patches meeting at the corner of Figure 17 cannot be GC^1 unless the three edges A, B, and C

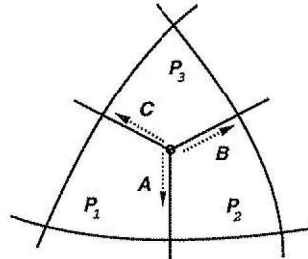


Fig 17. Three GC^1 patches must be coplanar at a common corner.

are coplanar; i.e., every incident edge is tangent, at the corner, to the plane defined by the other two edges.

At every corner, Surfplan determines which edges should be coplanar by performing a cluster analysis on the Gaussian sphere (i.e., a sphere of unit radius centered on the corner) using as points on the sphere the unit cross products of successive pairs of tangents at the corner. (Note the implied use of the canonical ordering.) For each cluster, a suitably averaged point is chosen, thus defining a common plane. All edges are corrected to lie on their common planes.

Experience at General Motors has shown that, even without enforcing coplanarity, most pairs of patches have an acceptable degree of GC^1 smoothness for numerically controlled (NC) machining. Since enforcing exact GC^1 smoothness raises patch degrees to 6, thereby slowing down numerical operations on patches, one possible strategy could be to record in the data structures all cases of approximate smoothness, while proceeding to exact smoothness only in the places where the current degree of smoothness is unacceptable. Such strategies have not been implemented.

4.7 Exact GC^1 Smoothness Using Higher Degrees

The final step for obtaining exact GC^1 smoothness via degree raising is described in [Sarraga 87, 90]. Unfortunately, in some cases the procedure introduces unwanted oscillations (i.e., bumpiness) in the final surface. Moreover, there is no simple way to characterize the corner and edge configurations that cause this problem. The problem may be an unavoidable drawback of all blending methods for surface construction. Nevertheless, in many cases, the degree-raising procedure produces excellent surfaces, even fixing ugly shapes in the vicinity of T-corners.

slowly than conventional languages like C. For cubic Bézier curves and untrimmed, low-degree Bézier patches, this limitation is not important. However, trimmed NURBS surfaces definitely require a faster language.

Within General Motors, Surfplan achieved its goal of automating important surfacing problems that could be handled with Bézier curves and surfaces and were consuming too much interaction time on conventional surfacing systems.

Appendix: Data Structures

This appendix outlines the basic data structures used by Surfplan to represent a surface model and its constituent parts, such as patches, edges, and corners. For greater clarity, the discussion omits or even changes the details actually implemented in code.

In the presentation below, names of entities are capitalized, and multiple words in a name are separated by underscores (“_”). Lisp syntax is used in the structure definitions, which have the form:

```
DEFSTRUCT <name-of-structure>  
  <list-of-elements>).
```

Readers acquainted with the C programming language may regard this pattern as essentially equivalent to:

```
typedef struct <name-of-structure>  
  { <list-of-elements> }  
<name-of-structure>;
```

Minor items, such as integer item counts and the like, are omitted below, because the purpose of this appendix is to present abstract connections between data items, and not to provide a sample piece of code.

A.1 The Surface

The top-level geometric data structure is the SURFACE, which groups together the numerical and topological components of a geometric model (typically a surface or a grid of curves on which a surface is to be constructed):

```
(DEFSTRUCT SURFACE  
  PATCHLIST  
  EDGELIST  
  CORNER_LIST  
  T_DEPENDENCY_GRAPH )
```

The PATCH_LIST, EDGE_LIST, and CORNER_LIST are unordered lists of surface components defined below. The T_DEPENDENCY_GRAPH lists the

acyclic directed graphs that record dependencies among edges in the vicinity of T-corners. (See Section 4.2.)

A.2 The Patch

The patch data structure contains both the numerical definition of the patch and the hooks to the topological information associated with the patch. Although only Bézier polynomials were implemented, the data structure scheme allows any numerical definition, since connectivity information is kept separate from numerical data.

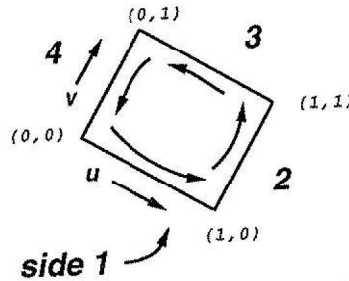


Fig 18. Conventions used for sides of patches.

Surfplan uses only untrimmed patches, although most modern surface modeling systems admit patches whose domain of definition has been trimmed by curves in the parameter domain. A good discussion of general trimming of parametric surfaces is found in [Farouki 87]. Associated numerical difficulties are outlined in [Sarraga & Waters 83].

```
(DEFSTRUCT PATCH
  NUMERICAL_DEFINITION
  SIDE_ARRAY
  OWNER_SURFACE )
```

The `SIDE_ARRAY` lists in order (counterclockwise around the sense of the patch normal) the data structures for the boundary sides of the patch. The patch is assumed to have distinct sides (four in Section 2), which are defined by the boundaries of the parameter domain. This assumption does not hold for trimmed patches, whose effective boundaries could be a single curve (e.g., a circle).

The sides of a `PATCH` are numbered 1-4 as one traverses them in counterclockwise order around the sense of the patch normal. The patch normal is

defined as the unit cross product of the partial derivative with respect to parameter u times the partial with respect to v . Denoting parametric coordinates by (u, v) , the starting corner is the point $(0, 0)$, so that side 1 has the range $(0 - 1, 0)$, side 2 has range $(1, 0 - 1)$, side 3 has range $(1 - 0, 1)$, and side 4 has range $(0, 1 - 0)$. (See Figure 18.)

The OWNER_SURFACE is a pointer to the SURFACE entity defined above. Such a reference is needed at the PATCH level when multiple instances of the SURFACE entity are handled by the software.

A.3 The Segment (on a Side of a Patch)

Each boundary side of a PATCH entity is a sequence of PATCH_SEGMENT entities (usually only one) that are ordered in the direction defined by the (u, v) ranges given above. The PATCH_SEGMENT entity defines, from the standpoint of the current PATCH, a curve segment at the boundary of the current PATCH. A patch side is segmented by T-corners, which allow the current PATCH to be adjacent to a PATCH that does not exactly match a side of the current PATCH (as in Figure 3).

Some information in the PATCH_SEGMENT entity is redundant, because it could be inferred from other data structures. However, experience has shown that this redundancy simplifies algorithms without introducing a significant programming overhead to ensure consistency.

```
(DEFSTRUCT PATCH_SEGMENT
  NUMERICAL_DEFINITION
  OWNER_PATCH
  OWNER_SIDE_N
  PREVIOUS_CORNER
  PREVIOUS_SEGMENT
  NEXT_SEGMENT
  ADJACENT_SEGMENT
  EDGE_DATA )
```

The NUMERICAL_DEFINITION entity defines the PATCH_SEGMENT with respect to its owning PATCH. For untrimmed Bézier patches, this definition consists only of (a) the parameter (u or v) that is constant along the current boundary side, and (b) the start and end values of the segment.

The elements OWNER_PATCH and OWNER_SIDE_N provide a path to the owning PATCH, which is used by algorithms that arrive at a PATCH_SEGMENT from a different connection, e.g., from the PATCH_SEGMENT of an adjacent PATCH.

Elements PREVIOUS_CORNER, PREVIOUS_SEGMENT, and NEXT_SEGMENT are redundant and simply provide adjacency information without having to ascend to the PATCH level.

The entity ADJACENT_SEGMENT denotes a matching PATCH_SEGMENT on an adjacent PATCH, if any. The matching PATCH_SEGMENT describes, in the parameter domain of the adjacent PATCH, the same space curve denoted by the current instance of PATCH_SEGMENT.

The entity EDGE_DATA provides a patch-independent parametrization of the space curve denoted by the current PATCH_SEGMENT. The curve parametrization generally exists before patches are created, i.e., when curves are created as explained in Section 4.1.

Entities NUMERICAL_DEFINITION, ADJACENT_SEGMENT (if not null), and EDGE_DATA provide alternative numerical definitions for the same geometric item, a segment on a space curve.

A.4 The Edge

The EDGE entity describes a segment of a space curve. An EDGE's NUMERICAL_DEFINITION belongs exclusively to the curve and is independent of adjacent patches, if these exist.

In addition, the EDGE entity is a convenient repository for numerical and topological information related to adjacent patches. (In most applications, almost every EDGE has at least one adjacent PATCH_SEGMENT.)

```
(DEFSTRUCT EDGE
  NUMERICAL_DEFINITION
  CORNER_LINKS
  INTERVAL_LINK
  PARALLEL_SEGMENT
  OPPOSITE_SEGMENT
  SENSE_OF_NORMALS
  BOUNDARY_DATA
  OWNER_SURFACE )
```

Every EDGE is bordered by two CORNER entities (defined below), which denote space points where several edges meet. An EDGE and a CORNER are always connected through a CORNER_LINK, described below.

The INTERVAL_LINK points to the space curve for which the current EDGE is a segment. If the current EDGE defines a complete space curve, this entity is null. The PARALLEL_SEGMENT is the PATCH_SEGMENT (possibly null) whose direction in its PATCH coincides with the direction assigned to the curve

segment represented by the EDGE. The OPPOSITE_SEGMENT corresponds to the opposite direction. After a surface is oriented (Section 4.4), every EDGE bordered by two patches has both parallel and opposite adjacent segments. The SENSE_OF_NORMALS is used during the algorithm for orienting the surface. The BOUNDARY_DATA entity stores continuity information between patches (e.g., whether adjacent patches are GC^1 or merely GC^0), while OWNER_SURFACE identifies the owning model.

A.5 The Corner

A CORNER denotes a point in R^3 that is an endpoint of one or more EDGE entities. Corners of patches (i.e., points where patch sides meet) always coincide with CORNER entities.

EDGE entities incident on a CORNER are listed in canonical order, i.e., in counterclockwise order around the sense of the surface normal at the corner point, with the first edge of the list selected arbitrarily. Because there may be more than one surface normal at a CORNER (i.e., when a sharp edge passes through the corner), the word "sense" is really an allusion to the surface orientation defined earlier.

An EDGE is connected to a CORNER only via the CORNER_LINK described below, which also stores endpoint-dependent data associated with an EDGE. The ARRAY_OF_LINKS lists the incident entities.

```
(DEFSTRUCT CORNER
  ARRAY_OF_LINKS
  CORNER_DATA
  OWNER_SURFACE )
```

A.6 The Corner-Data

The CORNER_DATA holds the coordinates of the CORNER point and, if the CORNER has a unique surface normal, the unit normal vector. If the CORNER has more than one surface normal, the corner is divided into subcorners, each of which has (a) a unique surface normal and (b) a list of edges that lie on the tangency plane denoted by the subcorner. Subcorners and edges within subcorners are listed in the canonical order of the surface.

Every CORNER entity (including subcorners) owns a CORNER_DATA. Therefore, the CORNER_DATA has mutually exclusive elements that are used depending on whether the owning corner is a subcorner or not.

```
(DEFSTRUCT CORNER_DATA
  POINT
  NORMAL
  IS_INTERIOR
  SUBCORNERS
  OWNING_CORNER
  NEXT_SUBCORNER )
```

IS_INTERIOR denotes whether a top-level corner (not a subcorner) is totally enclosed by patches. The entity SUBCORNERS lists all subcorners of a top-level corner, if any. The entities OWNING_CORNER and NEXT_SUBCORNER apply only to subcorners. OWNING_CORNER is a reference to the owning top-level CORNER, and NEXT_SUBCORNER provides immediate access to the next subcorner in the canonical ordering.

A.7 The Corner-Link (of an Edge)

The CORNER_LINK is a linking data structure between an EDGE and a CORNER. The CORNER_LINK functions as a facilitator for a number of access queries that occur often in algorithms dealing with corners of edges. For example, there are queries such as: "List, in the surface's canonical order, all the patches incident on the corner." The answer would be: "Traverse in order the list of CORNER_LINKs for the corner, pulling out the patch that owns the PATCH_SEGMENT attached to each CORNER_LINK."

```
(DEFSTRUCT CORNER_LINK
  OWNING_CORNER
  OWNING_EDGE
  END_NUMBER
  PREVIOUS_LINK
  NEXT_LINK
  OTHER_LINK
  PATCH_SEGMENT
  ENDPOINT_DATA )
```

The entities OWNING_CORNER and OWNING_EDGE are the linked CORNER and EDGE. The END_NUMBER specifies which end of an EDGE pertains to the CORNER_LINK. PREVIOUS_LINK and NEXT_LINK denote adjacent EDGE entities in the canonical order around a CORNER. OTHER_LINK refers to the CORNER_LINK at the opposite end of the owning EDGE. PATCH_SEGMENT denotes the patch following the owning edge in the canonical order around the owning CORNER.

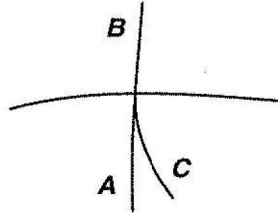


Fig 19. Edge *A* is tangent to *B* and cusps with *C*.

The `ENDPOINT_DATA` is a repository of endpoint conditions on the curve segment defined by the `EDGE` that owns the current `CORNER_LINK`. Two types of information are recorded: (a) tangency information relating to the edge, or edges, that continue the current edge across the corner, or meet it in a cusp from the same direction, as shown in Figure 19, and (b) coplanarity data at the current corner for two patches that are bordered by the current edge and the two following edges (in canonical order).

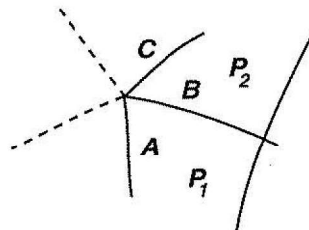


Fig 20. Edges *A*, *B*, and *C* are coplanar if patches P_1 and P_2 are GC^1 .

Before ensuring smoothness of unit normals across a boundary between two patches, the three patch boundaries meeting at each corner must be made coplanar. (See Figure 20.) In addition, patch smoothing algorithms need to know how the three corner tangents of these boundaries are related to each other.

The Surfplan implementation has not fully exploited the cusp and coplanarity information that can be stored in `ENDPOINT_DATA`.

A.8 The Segmented Curve

The SEGMENTED_CURVE entity defines a curve that spans more than one EDGE entity. Typically, curve segmentation is caused by T-corners (Section 4.2).

```
(DEFSTRUCT SEGMENTED_CURVE
  SUBCURVES
  CORNER_LINKS
  NUMERICAL_DEFINITION )
```

The SUBCURVES element references, in order, the INTERVAL_LINK entities defined below, which contain the EDGE entities comprising the segmented curve. The CORNER_LINKS element lists the end links of the complete curve for ease of access. The NUMERICAL_DEFINITION is typically a parametric Bézier curve, which is segmented by subdivision (Section 3).

A.9 The Interval Link

The INTERVAL_LINK entity links a SEGMENTED_CURVE to one of the contained EDGE entities. The INTERVAL_LINK also contains the numerical definition of an EDGE segment in terms of the numerical definition of the complete curve.

```
(DEFSTRUCT INTERVAL_LINK
  OWNING_CURVE
  EDGE
  NUMERICAL_DEFINITION )
```

OWNING_CURVE points to the owning SEGMENTED_CURVE, while the element NUMERICAL_DEFINITION typically contains the start and end parameter values for the current segment in the domain interval of the complete curve.

Some auxiliary elements have been omitted from the definition of this data entity and of the others presented above.

REFERENCES

[Bartels et al. 88]

Bartels, R.H., Beatty, J.C., and Barsky, B.A. (1987), *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann, Palo Alto, California.

[Bézier 86]

Bézier, P.E. (1986), *The Mathematical Basis of the UNISURF CAD System*, Butterworths, London.

[Boehm 80, 81]

Boehm, W. (1980), "Inserting new knots into B-spline curves," *Computer Aided Design*, Vol. 12, pp. 199-201; and Boehm, W. (1981), "Generating the Bézier points of B-spline curves and surfaces," *Computer Aided Design*, Vol. 13, pp. 365-366.

[Cavendish & Marin 92]

Cavendish, J.C., and Marin, S.P. (1992), "Feature-Based Surface Design and Machining," *IEEE Computer Graphics and Applications*, Vol. 12, No. 5, pp. 61-68, September 1992.

[Deo 74]

Deo, N. (1974), *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Englewood Cliffs, New Jersey.

[Farin 93]

Farin, G. (1993), *Curves and Surfaces for Computer Aided Design, A Practical Guide*, Academic Press, Cambridge, Massachusetts.

[Farouki 87]

Farouki, R.T. (1987), Trimmed-surface algorithms for the evaluation and interrogation of solid boundary representations, *IBM Journal of Research and Development* 31, 314-333.

[Faux et al. 79]

Faux, I.D., and Pratt, M.J. (1979), *Computational Geometry for Design and Manufacture*, Wiley, New York.

[Hoschek et al. 93]

Hoschek, J. and Lasser, D. (1993), *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley, Massachusetts.

[IGES 86]

Smith, B. and Wellington, J. (1986), *Initial Graphics Exchange Specification, Version 3.0*, , Publication NBSIR 86-3359, National Bureau of Standards, Gaithersburg, MD 20899.

[Oetjens et al. 90]

Oetjens, T.J., Chin, J.G., Ericson, B.J., and Sarraga, R.F. (1990), *Fast Computerization of Automotive Panels at General Motors*, General Motors Research Publication GMR-7100.

[Peters 91]

Peters, J. (1991), "Smooth Interpolation of a Mesh of Curves," *Constructive Approximation*, Vol. 7, pp. 221-246.

[Sarraga & Waters 83]

Sarraga, R.F., and Waters, W.C. (1983), "Free-Form Surfaces in GMSolid: Goals and Issues," in *Solid Modeling by Computers: From Theory to Applications*, M.S. Pickett and J.W. Boyse, eds., Plenum, New York, pp. 187-204.

[Sarraga 87]

Sarraga, R.F. (1987), " G^1 Interpolation of generally unrestricted cubic Bézier curves," *Computer Aided Geometric Design*, Vol. 4, No. 1, pp. 23-39; see also Sarraga, R.F. (1988), "Errata: G^1 Interpolation of generally unrestricted cubic Bézier curves," *Computer Aided Geometric Design*, Vol. 6, No. 2, pp. 167-171.

[Sarraga 90]

Sarraga, R.F. (1990), "Computer Modeling of Surfaces with Arbitrary Shapes," *IEEE Computer Graphics and Applications*, Vol. 10, No. 2, pp. 67-77, March 1990.

[Steele 84]

Steele, Guy L. (1984), *Common Lisp, The Language*, Digital Press, Burlington, Massachusetts.

[Veenman 82]

Veenman, P. (1982), The design of sculptured surfaces using recursive subdivision techniques, in: *Proc. Conf. on CAD/CAM Technology in Mechanical Engineering*, M.I.T., Cambridge, Massachusetts, pp. 54-63.

FORMER NOS ÉTUDIANTS À LA RECHERCHE

CLAUDE BREZINSKI

LABORATOIRE D'ANALYSE NUMÉRIQUE ET D'OPTIMISATION

UFR IEEA - M3

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

59655 - VILLENEUVE D'ASCQ CEDEX, FRANCE

e-mail: brezinsk@omega.univ-lille1.fr

Je n'ai pas la prétention, dans ce court article, de tout dire sur la façon de former nos étudiants à la recherche et je ne passerai en revue que certains aspects de la question. Des développements plus complets peuvent se trouver dans la vaste littérature sur ce sujet; en particulier, on pourra lire [1] où j'expose plus largement mes idées.

Pour faire de la recherche scientifique, dans quelque domaine que ce soit, il faut de l'intuition et de l'imagination. Ces deux notions étant voisines, précisons d'abord nos définitions. L'intuition est la certitude instinctive et immédiate de la vérité, sans recours au raisonnement ou à l'expérience. Des synonymes sont inspiration, instinct, illumination. L'imagination est la capacité d'élaborer des conceptions nouvelles et de trouver des solutions originales à des problèmes théoriques ou pratiques. Des synonymes sont originalité, inventivité, créativité. L'intuition vient du plus profond de l'individu et l'on peut être intuitif sans être imaginaire. L'intuition est inexplicable car c'est un éclair de la pensée et je pense donc qu'elle est innée et ne peut ni s'acquérir ni se développer. On peut seulement essayer de favoriser son apparition. Par contre, l'imagination procède plus d'une réflexion consciente, d'un travail interne et elle peut être développée. L'intuition sert l'imagination mais il faut ensuite de l'imagination pour asseoir sur des bases logiques ce qui avait été perçu dans un éclair. On trouvera dans [2] une discussion approfondie de ces concepts ainsi que des notions voisines.

Il me semble que le mode de pensée n'est pas le même chez le mathématicien que chez un chercheur en sciences de la nature (physique, chimie, biologie, etc...) tout simplement parce que celui-ci peut prendre appui sur une réalité tangible, visible, observable, ce qui est rarement le cas du mathématicien. Celui-ci, même s'il est un mathématicien appliqué réalisant des calculs sur ordinateur, a le plus souvent à démontrer des résultats théoriques, des théorèmes. Or, avoir l'intuition d'un théorème, pressentir qu'un certain résultat théorique pourrait être vrai est chose difficile. Comment, en effet, sans le secours d'aucune observation, d'aucun phénomène visible, pouvoir imaginer l'existence et la véracité

d'un théorème ? Comment penser à cela ? Je crois que personne n'est capable de répondre à une telle question car, sinon, cela signifierait qu'il existe des recettes pour faire de la recherche et qu'il suffit de les appliquer automatiquement. J'ai raconté, dans mon livre [1], comment se sont effectuées un certain nombre de découvertes et même de découvertes mathématiques (voir aussi [2, 5, 6, 7]). Mais cela ne mène certainement pas à une "algorithmique" de la recherche, de la découverte, de l'intuition, de l'inspiration, de l'idée novatrice et créatrice. Cela est d'autant plus vrai dans un domaine théorique comme les mathématiques, où le chercheur ne peut confronter directement ses idées théoriques à la réalité de l'expérience (au moins dans certains domaines des mathématiques). D'ailleurs nombre de chercheurs ont parlé de la fameuse "illumination" sans, pour autant, que cela illumine le lecteur sur le processus créatif.

Cela étant posé, on voit bien que l'intuition ne peut pas s'enseigner. Mais, cependant, ce n'est pas une raison pour laisser nos étudiants de thèse à l'abandon car, par contre, l'imagination peut et doit être cultivée et il est possible de la développer. Avoir de l'intuition et de l'imagination n'a rien à voir avec le fait de posséder ou non une culture encyclopédique dans son domaine. Ce sont deux choses de nature totalement différente et combien de fois avons nous rencontré de brillants agrégés totalement incapables de faire jaillir de leurs cerveaux la moindre idée novatrice et, inversement, combien d'étudiants bien moins érudits se sont révélés d'excellents chercheurs.

Examinons maintenant le cas des mathématiques. Je pense qu'il est très important, voire tout à fait fondamental, que de futurs chercheurs en mathématiques aient reçu un minimum de formation dans certaines sciences de la nature (physique ou mécanique en particulier) et que cela peut développer leurs facultés d'imagination. En effet, en physique, on observe un phénomène puis on essaye de l'expliquer c'est-à-dire d'en tirer des conclusions théoriques. On acquiert alors une gymnastique de l'esprit qui est très formatrice. Je suis certain que la pratique de la physique est indispensable au mathématicien, que l'intuition en physique, basée sur l'observation, aiguise l'esprit du futur mathématicien et le met en condition pour effectuer des découvertes. Autrefois, dans l'enseignement des mathématiques, c'était la géométrie qui jouait ce rôle formateur. Elle a, de façon malencontreuse, été presque totalement supprimée. Un exemple bien connu de cette influence réciproque entre mathématique et physique est la découverte des fractals par Benoit Mandelbrot grâce à la question de la longueur des côtes de la Bretagne. Notons que l'un de ses précurseurs dans cette étude fut le météorologiste Lewis Fry Richardson (celui de la méthode d'extrapolation).

On pensera aussi à la théorie du potentiel issue des problèmes d'attraction gravitationnelle et électrique. Dans un entretien, paru dans [3], Stanislaw M. Ulam parle des similarités de pensée entre un mathématicien et un physicien. Le mathématicien part de certains faits, qu'il appelle axiomes, et en déduit des conséquences qu'il appelle théorèmes. En physique on procède dans le sens inverse. Le physicien dispose d'un nombre important de faits, de relations, d'expressions formelles qui sont le résultat des expériences. Ensuite il cherche des lois simples, qui correspondent aux axiomes du mathématicien, dont il peut déduire ses résultats expérimentaux, qui correspondent aux théorèmes. Malgré cette différence, Ulam est convaincu que les modes de pensée et les intuitions ont une grande ressemblance dans les deux disciplines.

Nous devons aussi apprendre à nos étudiants à réfléchir. Je pense qu'il est inutile, et souvent même néfaste, que, dans les cours qui préparent directement à la recherche (les cours de DEA en France), les enseignants cherchent à gaver les étudiants de connaissances, de théorèmes et de démonstrations. Bien sûr, de tels cours sont faits pour amener les étudiants au niveau où s'effectue la recherche mondiale dans leur domaine. Cependant, je crois qu'il vaut mieux, comme le disait Pascal, "une tête bien faite qu'une tête bien pleine". Il est préférable d'apprendre moins de choses mais de les comprendre en profondeur, d'en comprendre le pourquoi et le comment. Il ne faut pas noyer nos étudiants sous un flot de détails techniques et de démonstrations qu'ils peuvent très bien, à ce niveau, aller voir eux-mêmes dans la littérature spécialisée. Il faut plutôt essayer de leur former l'esprit et de leur donner une vue d'ensemble, une vue synthétique du sujet en leur montrant comment les pièces du puzzle s'ajustent les unes aux autres, comment s'enchaînent les idées. En général les cours de ce niveau sont faits par des enseignants qui ont eux-mêmes contribué aux progrès dans le domaine enseigné. N'ayons pas de fausse modestie et parlons leur de la façon dont nous avons obtenu nous-mêmes nos propres résultats, de la façon dont les idées nous sont venues, comment elles se sont forgées puis reliées les unes aux autres. Expliquons leur ce qui a guidé notre démarche, racontons leur le cheminement de nos idées et les analogies que nous avons utilisées. N'ayons pas honte, non plus, de leur parler de nos erreurs et des voies sans issue dans lesquelles nous nous sommes aventurés car, à condition de savoir en tirer les bonnes conclusions, elles peuvent devenir une source d'inspiration. A mon avis, il est primordial que de futurs chercheurs connaissent la genèse d'un certain nombre de découvertes afin de démonter, tant que faire se peut, le mécanisme qui conduit à la découverte. Il faut leur faire comprendre que l'esprit travaille

souvent par analogie, en utilisant des idées provenant d'un autre problème. Par exemple, j'ai découvert le θ -algorithme (une méthode pour accélérer la convergence de certaines suites de nombres) en introduisant dans un autre algorithme d'accélération un paramètre dont j'ai ensuite cherché la valeur optimale. C'est-à-dire que je me suis directement inspiré du chemin tracé par David M. Young pour obtenir la méthode de sur-relaxation (SOR) à partir de la méthode de Jacobi pour résoudre les systèmes d'équations linéaires. George B. Dantzig raconte dans [4] comment il a été amené à la découverte de la méthode du simplexe. Ne pouvant trouver cet algorithme, il exposa à John von Neumann en quoi consistait la programmation mathématique. Immédiatement, von Neumann se mit au tableau et lui en fournit une théorie mathématique complète. Devant son ébahissement, il expliqua alors à Dantzig qu'il venait d'écrire avec Oskar Morgenstern un livre sur la théorie des jeux et, qu'en l'entendant, il avait immédiatement vu que les deux problèmes étaient équivalents. La théorie qu'il venait de lui présenter au tableau n'était que la transposition de son propre travail dans le langage de Dantzig. Mais, naturellement, tout le monde n'est pas John von Neumann ! Je suis certain que les récits de telles découvertes peuvent aider les étudiants et leur permettre de se forger peu à peu une méthodologie de la pensée, avec un certain nombre de petites astuces simples. Même si ce ne sont pas des recettes infaillibles pour découvrir quelque chose de nouveau, elles pourront au moins leur permettre d'éviter un certain nombre de pièges et leur feront peut-être gagner du temps. L'histoire de Dantzig leur montrera, pour le moins, qu'il faut savoir "penser à côté" selon la célèbre formule (de Pasteur, je crois). Souvent, quand on travaille sur une certaine question, c'est la solution d'un autre problème que l'on trouve, par analogie, par transport d'idées. Le mécanicien allemand Ernst Mach professait que l'on ne pouvait comprendre une théorie à moins de savoir comment elle avait été découverte (sans doute ne mit-il pas ses idées en pratique puisqu'il fut toujours opposé à la relativité restreinte d'Einstein). Naturellement, les étudiants peuvent (et doivent, pour leur culture générale trop souvent négligée au profit des seules lectures scientifiques liées à leurs recherches) lire des livres dans lesquels des chercheurs de diverses disciplines racontent leurs travaux. Mais rien ne vaudra jamais pour eux le contact direct avec des chercheurs plus âgés et nous devons les faire bénéficier de tout ce qui constitue notre propre expérience.

Il faut aussi apprendre à nos jeunes chercheurs que, pour obtenir des résultats nouveaux, il faut laisser vagabonder son esprit, le laisser aller là où il le veut, le laisser libre de toute entrave. Ainsi, avant de s'attaquer à un sujet, le

mathématicien français Paul Lévy ne lisait presque rien de ce qui avait déjà été fait. Cela, disait-il, orientait implicitement sa pensée dans une certaine direction et l'empêchait d'avoir des idées vraiment novatrices. Il en était de même de Chebyshev qui affirmait que trop de zèle à étudier les travaux des autres nuisait à l'originalité de ses propres travaux. Bien sûr, il ne faut pas tomber dans l'excès inverse et ne rien savoir de son sujet car l'on risque alors de n'enfoncer que des portes ouvertes. C'est là qu'intervient encore l'expérience de l'enseignant qui doit guider l'étudiant dans le choix de ses lectures. Dans le même ordre d'idée, lorsque l'on butte sur un obstacle que l'on n'arrive pas à franchir, il est bon de ne pas trop insister. Il vaut mieux s'arrêter et laisser l'esprit se reposer. Quand on reprend la question quelque temps après, on a oublié les fausses pistes sur lesquelles on s'était engagé, on repart avec un regard neuf sur la question et, en général, on la résoud (bien sûr, de nouveau, cela n'est pas une recette miracle !) car on ne reprend pas les chemins qui avaient conduit à une impasse. On s'était trompé de route pour atteindre le sommet et, comme l'alpiniste en difficulté, il faut avoir la sagesse de redescendre et de repartir par une autre voie au lieu de s'entêter et d'aller ainsi à l'échec et au découragement. Einstein travaillait sur un sujet jusqu'à saturation puis passait à autre chose ; c'est alors que l'idée qu'il attendait lui venait tout à coup. Il faut aussi savoir se reposer car c'est souvent dans les périodes d'inactivité intellectuelle apparente que la solution surgit sans que l'on s'y attende. Les cas d'illumination au réveil sont bien connus. Les idées peuvent également se présenter spontanément lorsque l'esprit est totalement au repos ou qu'il se livre à une activité différente. De très nombreux cas sont décrits dans [1, 5, 6, 7].

Il faut aussi dire à nos étudiants que, pour trouver un résultat général, il est souvent nécessaire de commencer par réfléchir sur un grand nombre de cas particuliers pour essayer de dégager leurs caractéristiques communes, leurs propriétés communes, de voir leurs ressemblances et aussi leurs différences afin d'en déduire le cas général. Hermite disait qu'il se transformait en naturaliste et que c'était l'observation attentive des faits qui était la source la plus féconde des découvertes mathématiques [8]. "L'analyse numérique est, pour beaucoup, une science expérimentale" comme l'a écrit Peter Wynn dans l'un de ses articles. Il me semble que la première chose à faire, lorsque l'on découvre un nouvel algorithme, est de le programmer et de l'essayer sur plusieurs exemples numériques avant de se lancer dans une quelconque démonstration. C'est, en effet, inutile de chercher une preuve de convergence si le premier exemple pratique ne converge pas. Mais, comme je l'ai déjà dit plus haut, il faut aussi savoir exploiter une

erreur et se demander alors pourquoi cet exemple particulier n'a pas marché ou ce qui, dans l'algorithme, n'est pas au point. En analyse numérique on peut ainsi évoluer par aller-retours entre théorie et expérience, exactement comme dans les sciences de la nature.

Il est également important pour nos étudiants d'aller régulièrement à la bibliothèque pour, bien sûr, consulter les articles dont ils ont besoin mais aussi pour butiner (selon la traduction de l'anglais "to browse" donnée par le dictionnaire et qui me paraît très bien adaptée à ce genre d'activité) au hasard dans les livres et les journaux. On tombe quelquefois en arrêt, comme un chien de chasse lorsqu'il lève un gibier, devant une formule qui, même si elle n'est pas directement utilisable dans son propre travail, peut déclencher l'intuition, l'étincelle qui aboutira au succès ou, tout au moins, mettra sur la voie. C'est ce que je préconise à mes étudiants; on pourra lire, à ce sujet, l'entretien avec Persi Diaconis dans [3]. En général, cela se produit dans un article qui ne vous aurait attiré ni par son titre ni même par son résumé. Je crois beaucoup, pour des raisons économiques évidentes, au développement rapide des journaux électroniques dans un futur proche. Malheureusement, la pratique du butinement risque d'en être sérieusement affectée et, par là, une source importante d'inspiration et d'intuition risque de se tarir. Puisque je parle de bibliothèque, nous devons également apprendre à nos étudiants à s'en servir de façon efficace. En effet, combien de temps perdent-ils, au début de leur thèse, à s'y orienter, à faire connaissance avec les journaux qui publient des analyses d'articles classés par thèmes, à faire la bibliographie du sujet qui leur a été confié, etc... Cela fait aussi partie de la formation que nous nous devons de leur donner.

Il y a encore beaucoup d'autres choses importantes à dire à nos étudiants pour les former à la recherche. Par exemple, il faut leur dire qu'il est difficile d'avoir des idées simples, que ce sont d'ailleurs les plus élégantes et qu'elles ne paraissent simples qu'après avoir été trouvées et débarrassées de tout ce qui était inutile. Malheureusement, certains auteurs pensent que plus un article est compliqué et ardu à comprendre plus il a de valeur car il met mieux en relief leurs connaissances et les difficultés qu'ils ont dû surmonter. Bien au contraire, c'est la simplicité qui confère sa beauté à un résultat et c'est d'ailleurs l'effort de simplification qui est souvent le plus difficile. Il faut leur dire que l'on se demande souvent, devant un résultat simple que l'on vient d'obtenir, pourquoi personne n'y avait pensé plus tôt. Mais il en est ainsi; ils doivent l'admettre et surtout se réjouir de l'avoir trouvé. Il faut leur dire que l'on fait parfois un long détour pour arriver à un résultat tout proche et qu'il faut alors en chercher

une approche plus directe (par exemple, la première démonstration par Hermite de la transcendance du nombre e comportait quelques 73 pages alors que les démonstrations modernes sont très courtes). Nous devons également aider nos étudiants à présenter leurs résultats aussi bien par écrit dans un article ou sur des transparents que par oral lors d'un séminaire ou d'un congrès. Il faut les pousser à rédiger leurs résultats dans leur forme presque définitive au fur et à mesure qu'ils les obtiennent. Nous devons leur faire savoir ce qu'est la vie d'un scientifique, ce qu'est la vie académique, leur faire faire connaissance avec les dédales à suivre pour publier un article, et bien d'autres choses encore. Naturellement, je sais très bien que je viens de décrire la formation idéale que l'étudiant devrait recevoir et que cela ne se passe malheureusement pas toujours ainsi dans la réalité.

Diriger la thèse d'un étudiant implique, de notre part, beaucoup plus que lui fournir un sujet, suivre ses travaux et l'aider dans les difficultés techniques de certaines démonstrations. Nous devons lui apprendre ce qu'est la recherche, ce qu'est le métier de chercheur. Il faut lui dire que la voie de la découverte est souvent longue, difficile et solitaire. Il faut lui apprendre à ne surtout jamais se décourager même lorsque l'on trouve un résultat déjà connu. C'est alors la preuve que l'on est dans la bonne voie et il faut persévérer. Il faut lui faire comprendre que des résultats négatifs ou des contre-exemples peuvent être aussi intéressants et importants que des résultats positifs à condition de savoir en tirer les bonnes conclusions, la "substantifique moelle" chère à Rabelais. C'est à nous de l'aider, c'est à nous de l'encourager dès qu'il obtient seul ses premiers résultats, c'est à nous de le guider vers les joies de la découverte.

Bibliographie

- [1] C. Brezinski, *El Oficio de Investigador*, Siglo XXI, Madrid, 1993.
- [2] A.F. Osborn, *Créativité, l'Imagination Constructive*, Dunod, Paris, 1988.
- [3] D.J. Albers, G.L. Alexanderson, eds., *Mathematical People*, Birkhäuser, Boston, 1985.
- [4] D.J. Albers, G.L. Alexanderson, C. Reid, eds., *More Mathematical People*, Harcourt Brace Jovanovich, Boston, 1990.
- [5] A. Kohn, *Par Hasard ou par Erreur ?*, Eshel, Paris, 1990.
- [6] R.M. Roberts, *Serendipity, Accidental Discoveries in Science*, Wiley, New York, 1989.
- [7] F. Vidal, *L'Instant Créatif*, Flammarion, Paris, 1984.
- [8] C. Brezinski, Charles Hermite, père de l'analyse mathématique moderne, *Cahiers Hist. Phil. Sci.*, 32 (1990) 1-81.

INFORMACIÓN SOBRE LA NUEVA REVISTA ELECTRÓNICA "ESAIM: COCV"

La Sociedad Francesa SMAI ha decidido lanzar una nueva serie de revistas electrónicas, denominada ESAIM (European Series in Applied and Industrial Mathematics) y en particular la revista

ESAIM: Control, Optimisation and Calculus of Variations

o simplemente, ESAIM: COCV.

El objeto de esta revista es publicar artículos en las áreas del Control, la Optimización y el Cálculo de Variaciones, tanto en lo que se refiere a la teoría como a sus aplicaciones. Esta revista tiene como objetivo alcanzar el nivel de las mejores revistas del área tanto en lo que se refiere a sus contenidos como a la exposición y presentación. Obviamente, algunas de las grandes ventajas que ésta presentará frente a las clásicas revistas impresas es la celeridad a la hora de publicar y difundir los artículos, la ilimitada difusión y accesibilidad desde cualquier lugar del mundo y la posibilidad de publicar trabajos largos pues no se ha establecido limitación alguna en el número de páginas de cada artículo. Trabajos de carácter expositivo o surveys serán también considerados.

J. M. Coron (ENS, Cachan, Francia) es el Editor en Jefe de la revista que cuenta asimismo con tres Editores Adjuntos: J. Blum (Grenoble, Francia), J. P. Quadrat (INRIA, Francia) y E. Zuazua (UCM, España). El Comité Editorial cuenta con una treintena de especialistas internacionales entre los que cabe mencionar a nuestro colega E. Casas de la Universidad de Cantabria.

Los primeros artículos estarán disponibles en el centro de servicio electrónico a partir del mes de Septiembre aproximadamente. Mientras el servicio no esté instalado, los autores que estén interesados en enviar un artículo, pueden hacerlo a la dirección electrónica de la Secretaría abajo indicada.

Una vez que el artículo sea aceptado, será puesto a disposición pública en el servidor que será de acceso libre. Mediante login remoto se podrá obtener el fichero dvi o postscript del artículo, pero el fichero fuente no será accesible.

Regularmente se hará una publicación impresa de los resúmenes de los artículos (entre 1 y 3 páginas por artículo) y se producirán CD-ROMS con

el contenido total de la revista durante intervalos de tiempo adecuados. La suscripción a este material se mantendrá lo más barata posible.

En estos momentos se puede acceder al logotipo y diversas informaciones más detalladas sobre esta revista a través de [www](http://www.emath.fr//cocv/):

URL:<http://www.emath.fr//cocv/>

Aquellos interesados en mantenerse informados de esta iniciativa pueden ponerse en contacto por e-mail con la Secretaría Editorial de la revista:

C. PERRIGAULT
ESAIM:COCV
SMAI, IHP
11, rue Pierre et Marie Curie
F-75231 Paris Cedex 05.
Tel: 33-1-44 27 66 61
Fax: 33-1-44 07 03 64
e-mail:cocv@esaim.emath.fr

ENRIQUE ZUAZUA
UNIVERSIDAD COMPLUTENSE DE MADRID

PREMIO REAL ACADEMIA DE CIENCIAS

EXACTAS, FÍSICAS Y NATURALES 1995

A JESÚS M. SANZ SERNA

La Real Academia de Ciencias Exactas, Físicas y Naturales, en su sesión inaugural del curso 1995-96 el pasado mes de octubre, otorgó los Premios que llevan su nombre. Estos premios son concedidos, a propuesta de los Académicos Numerarios y Correspondientes Nacionales, a *autores españoles por su labor de investigación científica o técnica relacionada con las materias que la Academia cultiva y sus aplicaciones, debiendo ser una labor realizada en España que muestre la excelencia y la creatividad investigadora.*

Tenemos la satisfacción de comunicar que uno de los dos premios de este año ha sido concedido a nuestro compañero Jesús M. Sanz Serna. En el anterior número del Boletín le felicitábamos por la concesión de otros dos importantísimos premios y se recordaba su gran contribución al avance del Análisis Numérico, no sólo en España sino en el mundo, por lo que no repetiremos aquí un resumen de su obra hasta el momento actual. Sin embargo, sí cabe reiterar el

orgullo que supone para toda la Matemática Aplicada española (y en particular para SEMA, de la que ha sido Vicepresidente el año pasado) contar entre sus miembros con un investigador tan destacado. Afortunadamente en los últimos años han sido varios premios muy importantes los que han recaído en investigadores de nuestra área y Sociedad, lo que contribuye a elevar el prestigio de ellas y en consecuencia a beneficiarnos a todos.

MARIANO GASCA

XIV CEDYA/IV CMA

Durante los días 18 al 22 de septiembre de 1995 se celebró en Vic (Barcelona) una nueva edición del Congreso de Ecuaciones Diferenciales y Aplicaciones / Congreso de Matemática Aplicada. Dentro del marco de este congreso se incluyó la primera Jornada Industrial de SEMA.



En la foto se muestra la mesa presidencial durante la sesión inaugural del congreso. De izquierda a derecha: Gerard Gómez, secretario de SEMA (actualmente vicepresidente de la Sociedad); Antoni Giro, director general de investigación de la Generalitat catalana; Ricard Torrents, director general de los Estudios Universitarios de Vic; Carles Simó, presidente del Comité organizador del congreso; Santiago Benito, primer teniente alcalde de Vic.

Una especialidad de Matemática Aplicada en el nuevo plan de estudios de Matemáticas de la Universidad Complutense de Madrid

El pasado 24 de Abril de 1995 apareció publicado en el BOE el nuevo plan de estudios de la Licenciatura de Matemáticas de la Universidad Complutense de Madrid. Las novedades de este nuevo plan con respecto al antiguo son muchas pero una de ellas es de una especial relevancia en nuestro contexto: la aparición de una especialidad de Matemática Aplicada. El camino recorrido por todo el profesorado de los Departamentos adscritos a la Facultad de Matemáticas, sus órganos de gobierno y especialmente por los miembros de la Comisión de Planes de Estudios de la Facultad fue muy largo y laborioso. Sintonizar las muy distintas opiniones de un colectivo tan numeroso y complejo como el del profesorado de la Facultad con más de 250 docentes fue una difícil tarea. La Comisión comenzó sus trabajos ya en 1990 desarrollando durante el proceso de elaboración numerosas sesiones semanales de trabajo con las correspondientes consultas a los departamentos y todo ello independientemente de los cambios que se produjeron en ese largo período de tiempo (cambios de Decano, cambios de Directores de Departamento, renovación de la Junta de Facultad ,etc.). Solo tres personas, de las 12 que componían la Comisión, participaron en todo el largo proceso, figurando entre ellas quien escribe estas líneas.

El nuevo plan es pues fruto de las sintonías conseguidas y probablemente dista bastante de cada una de las expectativas iniciales los Departamentos pero tiene la virtud de hacer realidad una buena parte de aquellas expectativas. Con una carga total de 320 créditos, el plan de estudios consiste en dos ciclos de 3 y 2 años respectivamente. Todos los cursos son cuatrimestrales excepto el primero que tiene carácter anual. Las materias obligatorias de universidad están emplazadas únicamente en el primer ciclo y alcanzan un total de 57 créditos. Las materias troncales, de obligada inclusión, se distribuyen en 81 créditos en el primer ciclo y 46,5 en el segundo. Las materias optativas del primer ciclo suman un total de 30 créditos y las del segundo 73,5 créditos. Los alumnos pueden optar también a 32 créditos de libre configuración. Señalemos, así mismo que los alumnos pueden conseguir 15 créditos por practicas en empresas, instituciones públicas o privadas, etc, o bien por trabajos académicamente dirigidos.

El tema de la configuración de las especialidades fue uno de los más conflictivos. En especial lo fue la creación de la especialidad de Matemática Aplicada sustituyendo, en algún modo, a la antigua especialidad de Mecánica y que ahora deaparece. Las especialidades tienen un carácter abierto configurandose por medio de una serie de materias (entre 12 y 14) de las que el alumno debe elegir

las que le permitan alcanzar 75 créditos. En las hojas adjuntas se pormenorizan detalles.

Estamos seguros de que los años venideros mostraran que la especialidad de Matemática Aplicada obedece a una diferenciación, no solo temática si no también de salidas profesionales, con respecto a otras especialidades. Estamos convencidos de que innovaciones de este estilo benefician de hecho a todo el colectivo de matemáticos como así se ha mostrado ya en otros países.

JESÚS ILDEFONSO DÍAZ
UNIVERSIDAD COMPLUTENSE DE MADRID

**PLAN DE ESTUDIOS:
LICENCIADO EN CIENCIAS MATEMÁTICAS
UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS**

Aprobado por Resolución de 29 de marzo de 1995
(B.O.E. 24/04/95)

ESTRUCTURA Y CARGA LECTIVA

ESTRUCTURA: Primero y Segundo Ciclos

DURACIÓN: 3 años (I ciclo) + 2 años (II ciclo) = 5 años

CARGA LECTIVA: 186 créditos (I ciclo) + 134 créditos (II ciclo) = 320 créditos

**ASIGNATURAS
CRÉDITOS**

PRIMER CICLO

TRONCALES 81
OBLIGATORIAS 57
OPTATIVAS 30
LIBRE CONFIGURACIÓN 18

SEGUNDO CICLO

TRONCALES 46,5
OBLIGATORIAS 00,0
OPTATIVAS 73,5
LIBRE CONFIGURACIÓN 14,0

ITINERARIO (sin contar con los créditos de libre configuración)

1º	Álgebra Lineal y Geometría	18	Total
	Análisis de Variable Real	18	
	Informática	9	54
	Álgebra Básica	9	

PRIMER CUATRIMESTRE			SEGUNDO CUATRIMESTRE			
2º	Métodos Numéricos	10,5	Total	Geometría Proyectiva	9	Total
	Cálculo Diferencial	9		Cálculo Integral	7,5	
	Probabilidad y Estadística	10,5	Métodos Estadísticos	7,5	24	

3º	Elementos de Geometría diferencial y Topología	7,5	Total	Geometría Diferencial de Curvas y Superficies	9	Total
	Eltos. de Var. Compleja y Ec. Diferenciales	7,5		Investigación Operativa	7,5	
	Optativa	7,5	Optativa	7,5	31,5	
	Optativa	7,5	Optativa	7,5		

4º	Estructuras Algebraicas	9	Total	Variedades Diferenc.	6	Total
	Ec. Diferenciales	9		Variable Compleja y Análisis Funcional	9	
	Topología General	4,5	Optativa	7,5	30	
	Optativa	7,5	Optativa	7,5		

5º	Análisis Numérico de Ec. Diferenciales	9	Total	Optativa	7,5	Total
	Optativa	7,5		Optativa	7,5	
	Optativa	7,5	Optativa	7,5	30	
	Optativa	6	Optativa	7,5		

MATERIAS OPTATIVAS

ASIGNATURA	CR. Teóricos	CR. Prácticos	CR. Totales
Astronomía y Geodesia	4,5	3	7,5
Astronomía fundamental	4,5	3	7,5
Curvas algebraicas**	4,5	3	7,5
Historia de las Matemáticas I	6	1,5	7,5
Inferencia estadística**	4,5	3	7,5
Lógica matemática**	4,5	3	7,5
Mecánica clásica	4,5	3	7,5
Programación matemática I**	4,5	3	7,5
Teoría de la medida**	4,5	3	7,5
Teoría elemental de números	4,5	3	7,5

(**) Estas asignaturas podrán cursarse como optativas de Segundo Ciclo, si no se han cursado como optativas de Primer Ciclo o como libre configuración.

DISTRIBUCIÓN DE LOS CRÉDITOS:

Asignaturas TRONCALES	36 (21 teóricos + 15 prácticos)
Asignaturas OBLIGATORIAS	48 (30 teóricos + 18 prácticos)
Asignaturas OPTATIVAS	30 (***)
LIBRE CONFIGURACIÓN	10,5(***)

TOTAL 124,5 CRÉDITOS

(***) La distribución en créditos teóricos y prácticos dependerá de las materias elegidas.

SEGUNDO CICLO

CUARTO Y QUINTO CURSOS

ASIGNATURA y Tipo	CR. Teóricos	CR. Prácticos	CR. Totales
Análisis numérico de Ecuaciones diferenciales (TR)	6	3	9
Ecuaciones diferenciales (TR)	6	3	9
Estructuras algebraicas (TR)	6	3	9
Topología general (TR)	3	1,5	4,5
Variable compleja y Análisis funcional (TR)	6	3	9
Variedades diferenciables en el espacio euclídeo (TR)	4,5	1,5	6
Asignaturas optativas			73,5
Libre configuración			14

Observación: Deberán cursarse 73,5 créditos de materias optativas a elegir de entre las asignaturas siguientes que no se hayan cursado como optativas de Primer Ciclo o como libre configuración:

ASIGNATURA	CR. Teóricos	CR. Prácticos	CR. Totales
Álgebra computacional	4,5	3	7,5
Álgebra conmutativa	4,5	3	7,5
Álgebra homológica*	4,5	3	7,5
Análisis complejo	4,5	3	7,5
Análisis de Fourier	4,5	3	7,5
Análisis exploratorio de datos	3	1,5	4,5
Análisis funcional	4,5	3	7,5
Análisis numérico de ec. en derivadas parciales	4,5	3	7,5
Análisis real*	4,5	3	7,5
Astrometría	4,5	3	7,5
Calculabilidad y complejidad	6	3	9
Cartografía Y Fotogrametría	4,5	3	7,5
Curvas algebraicas	4,5	3	7,5

Diseño de experimentos	3	3	6
Ec. en derivadas parciales I	4,5	3	7,5
Ec. en derivadas parciales II	4,5	3	7,5
Estructuras de datos y algoritmos	7,5	4,5	12
Geodesia espacial	4,5	3	7,5
Geodesia física	4,5	3	7,5
Geodesia geométrica y gravimetría	4,5	3	7,5
Geometría algebraica y analítica	4,5	3	7,5
Geometría de variedades diferenciables	4,5	3	7,5
Geometría Riemanniana	4,5	3	7,5
Historia de las matemáticas II	6	1,5	7,5
Inferencia estadística	4,5	3	7,5
Laboratorio de programación	0	9	9
Lógica matemática	4,5	3	7,5
Mecánica celeste	4,5	3	7,5
Mecánica de medios continuos*	4,5	3	7,5
Métodos de regresión y análisis multivariante	4,5	3	7,5
Metodología matemática	4,5	3	7,5
Modelos de gestión y producción	4,5	1,5	6
Operadores y álgebras de Banach*	4,5	3	7,5
Optimización combinatoria y teoría de grafos	4,5	3	7,5
Probabilidad I	4,5	3	7,5
Procesos estocásticos	4,5	3	7,5
Programación matemática I	4,5	3	7,5
Programación matemática II	3	1,5	4,5
Relatividad y cosmología	4,5	3	7,5
Series temporales	3	1,5	4,5
Simulación	3	1,5	4,5
Sistemas dinámicos y teoría de control*	4,5	3	7,5
Teoría algebraica de números*	4,5	3	7,5
Teoría de autómatas y lenguajes formales	6	3	9
Teoría de conjuntos	4,5	3	7,5
Teoría de errores	4,5	3	7,5
Teoría de la medida	4,5	3	7,5
Teoría de la programación	6	3	9
Topología algebraica	4,5	3	7,5
Topología de variedades*	4,5	3	7,5
Topología diferencial*	4,5	3	7,5

(*) Asignaturas bienales. De estas 8 asignaturas, tan sólo se impartirán cuatro, en años alternativos.

PERFILES

Los siguientes grupos de asignaturas darán lugar a los perfiles que se mencionan:

1. Perfil de **Astronomía y Geodesia**

Astrometría – Astronomía Fundamental – Cartografía y Fotogrametría – Ecuaciones en Derivadas Parciales I – Geodesia Espacial – Geodesia Física – Geodesia Geométrica y Gravimetría – Mecánica Celeste – Mecánica Clásica – Teoría de Errores

2. Perfil de **Estadística e Investigación Operativa**

Análisis Exploratorio de Datos – Diseño de Experimentos – Inferencia Estadística – Métodos de Regresión y Análisis Multivariante – Modelos de Gestión y Producción – Optimización Combinatoria y Teoría de grafos – Probabilidad I – Procesos Estocásticos – Programación Matemática I – Programación Matemática II – Series Temporales – Simulación

3. Perfil de **Matemática Aplicada** (elegir 75 créditos entre:)

Álgebra Computacional – Análisis de Fourier – Análisis Funcional – Análisis Numérico de Ecuaciones en Derivadas Parciales – Ecuaciones en Derivadas Parciales I – Ecuaciones en Derivadas Parciales II – Geometría de Variedades Diferenciables – Mecánica Clásica – Mecánica de Medios Continuos – Programación Matemática I – Relatividad y Cosmología – Sistemas Dinámicos y Teoría de Control – Teoría de la Medida

4. Perfil de **Matemática Computacional** (elegir 75 créditos entre:)

Álgebra Computacional – Calculabilidad y Complejidad – Estructuras de Datos y Algoritmos – Laboratorio de Programación – Lógica Matemática – Optimización Combinatoria y Teoría de Grafos – Programación Matemática I – Simulación – Teoría de Autómatas y Lenguajes Formales – Teoría de la Programación

5. Perfil de **Matemática Fundamental** (elegir 75 créditos entre:)

Álgebra Conmutativa – Análisis Complejo – Análisis de Fourier – Análisis Funcional – Curvas Algebraicas – Ecuaciones en Derivadas Parciales I – Geometría Algebraica y Analítica – Geometría de Variedades Diferenciables – Geometría Riemanniana – Mecánica Clásica – Teoría de Conjuntos – Teoría de la Medida – Topología Algebraica

6. Perfil de **Metodología** (15 créditos en prácticas de enseñanza más 67,5 créditos a elegir entre:)

Astronomía Fundamental – Curvas Algebraicas – Estructuras de Datos y Algoritmos – Historia de las Matemáticas II – Inferencia Estadística – Mecánica Clásica – Metodología Matemática – Teoría de Conjuntos – Teoría de la Medida – Topología Algebraica

Control of Partial Differential Equations and Applications

Eduardo Casas

Editado por Marcel Dekker, Inc., 1995

320 páginas. ISBN 0-8247-9607-1

Este volumen contiene los textos de las conferencias impartidas en *IFIP TC7/WG-7.2 Conference on Control of Partial Differential Equations and Applications* que tuvo lugar en Laredo (Cantabria), durante los días 5-9 de Septiembre de 1994. Los tópicos que se discutieron fueron: controlabilidad y estabilización de sistemas, condiciones de optimalidad, métodos numéricos, y aplicaciones.

Free Boundary Problems: Theory and Applications

J.I. Díaz, M.A. Herrero, A. Liñán y J.L. Vázquez

Editado por Longman Scientific and Technical, 1995

219 páginas. ISBN 0-582-25645-3

El libro recoge algunas de las contribuciones de la conferencia internacional *Free Boundary Problems: Theory and Applications*, celebrada en Toledo, del 21 al 26 de Junio de 1993. Las sesiones de aquella conferencia fueron las siguientes: problemas de frontera libre en la industria europea, problemas de frontera libre en la industria de Estados Unidos de América, flujos de curvatura media, transiciones de fase y ciencias de los materiales, mecánica de fluidos, problemas de combustión, fronteras libres en problemas de evolución y control e identificación.

Fluid Mechanics. Lecture Notes of Summer Schools

Manuel G. Velarde

Editado por World Scientific Publishing Co., 1995

584 páginas. ISBN 981-02-1899-0.

El libro es una colección de contribuciones de varios cursos de verano sobre aspectos teóricos, experimentales y aplicaciones de las muchas ramas de la física de fluidos. Se describen interesantes propiedades de los fluidos no-Newtonianos, heterogéneos y macromoleculares. Temas tales como los balances entre la tensión superficial y convección así como los efectos de la tensión en el caso de superficies

no homogéneas (gotas, puentes líquidos, convección celular, ondas interfaciales tales como ondas de solitones y trenes de ondas, etc.) son el objeto de diversos capítulos. Fenómenos caóticos y algunas propiedades del comportamiento turbulento son también analizados.

Modelado de Sistemas en Oceanografía, Climatología y Ciencias Medio-Ambientales: Aspectos Matemáticos y Numéricos

Antonio Valle y Carlos Parés

Editado por Grupo de Análisis Matemático Aplicado de la Universidad de Málaga, 1994

239 páginas. ISBN 84-7496-269-2.

Este volumen contiene los textos de las conferencias y comunicaciones relativas al Simposio sobre "Modelado de Sistemas en Oceanografía, Climatología y Ciencias Medioambientales: Aspectos Matemáticos y Numéricos" que, organizado por el Grupo de Investigación de Análisis Matemático Aplicado de la Universidad de Málaga, tuvo lugar en el Hotel D. Pablo de Rorremolinos durante los días 24, 25 y 26 de Enero de 1994.

Memoria de Análisis Numérico

Juan M. Viaño Rey

Distribuido por Tórculo Artes Gráficas, S.A.L.

C./ Rosalía de Castro, 45 Bajo. 15706 Santiago de Compostela.

Tlf. (981)592600. Fax: (981) 591863

729 páginas. ISBN 84-605-2649-6.

Esta memoria de Análisis Numérico es una edición puesta al día del proyecto docente presentado por el autor para optar a la cátedra que actualmente ocupa en la Universidad de Santiago. En efecto, aprovechando la entrada en vigor del nuevo Plan de Estudios en la Licenciatura de Matemáticas de dicha Universidad se revisan los contenidos, enfoques y metodología de las materias de análisis numérico que en ella se imparten.

La memoria se divide en tres partes que contemplan, respectivamente, el desarrollo conceptual (604 páginas), la bibliografía general (50 páginas) y algunos comentarios sobre la especialidad de Matemática Aplicada y la enseñanza del Análisis Numérico en el Plan de Estudios de Matemáticas (70 páginas).

En la Primera Parte, titulada *Desarrollo Conceptual del Análisis Numérico*, se exponen los rasgos más destacados de cada uno de los temas principales del

análisis numérico. El orden de la exposición no coincide, necesariamente, con el que se sigue en las sucesivas disciplinas de la licenciatura sino que se realiza, dentro de cada tema, un tratamiento global de principio a fin independientemente de que el tema deba ser “repartido” en dos o más asignaturas distintas. La extensión y profundidad con que se tratan los temas es variable: mientras algunos resultados clásicos se exponen de forma muy concisa, otros de desarrollo reciente se tratan con mayor amplitud y detalle, incluyendo una panorámica general del mismo.

La Segunda Parte se titula *Bibliografía General de Análisis Numérico* e incluye una relación de fuentes bibliográficas distribuidas en los grupos siguientes: i) libros agrupados por su contenido temático; ii) relación de revistas y publicaciones de análisis numérico y temas afines; iii) soportes lógicos del “software” de métodos numéricos más importantes actualmente distribuidos.

En la Tercera Parte se hacen algunas consideraciones sobre *Matemática Aplicada y Análisis Numérico en la Licenciatura de Matemáticas*, se presenta lo que pretende ser “una primera iteración” de los programas de las diferentes asignaturas que constituyen el “cuerpo básico” del análisis numérico en dicha licenciatura y se acompañan algunos comentarios que intentan reflejar los criterios que han guiado este diseño y los aspectos pedagógicos que deben contemplarse a la hora de impartirlos.

(JUAN M. VIAÑO)

Modelo Lineal de Láminas Delgadas de Koiter: Fundamentación Matemática y Aproximación Mediante un Método no Conforme de Elementos Finitos

María Ángeles Vilarriño Moreno

Editado por Grupo de Análisis Matemático Aplicado de la Universidad de Málaga, 1995

239 páginas. ISBN 84-7496-293-5.

Este libro contiene la memoria de doctorado de la autora, realizada bajo la dirección del Dr. Francisco J. Palma. En el presente trabajo se profundiza en el estudio de la deformación de las láminas delgadas elásticas, concretamente en los siguientes aspectos: el análisis matemático de la obtención del modelo lineal de Koiter y el análisis numérico de la aproximación de la solución mediante un método no conforme de elementos finitos, el llamado método delincuente de Sander.

CONTENIDOS: 1 - Elementos de Geometría Diferencial. 2 - Deformación de Láminas Delgadas. 3 - Energía Potencial Total. 4 - El Problema Continuo. 5 - El Problema Discreto. 6 - El Método Delincuente de Sander. 7 - Resultados Numéricos.

EDUARDO CASAS

En esta sección del boletín recogeremos los libros publicados en el campo de la Matemática Aplicada. Por lo tanto, si sois autores de algún libro, aquí podéis encontrar un lugar para darlo a conocer. También será bienvenida información sobre novedades bibliográficas correspondientes a otros autores y que consideréis de gran interés. Los datos sobre los libros para su inclusión en esta sección podéis enviarlos por correo electrónico a nuestra dirección:

sema@masun1.unican.es

CONFERENCIA INTERNACIONAL SOBRE
PEQUEÑOS SATÉLITES:
MISIONES Y TECNOLOGÍA

Madrid, octubre de 1996

Organizado por el Instituto Nacional de Técnica Aeroespacial y la Organización de Naciones Unidas.
Patrocinado por la Agencia Europea del Espacio.

RESÚMENES DE TESIS

GENERACIÓN DE MALLAS TRIDIMENSIONALES
MEDIANTE LA TRIANGULACIÓN DE DELAUNAY

Doctorando: José María Escobar Sánchez.

Director: Rafael Montenegro Armas.

Defensa: 31 de marzo de 1995, Universidad de las Palmas de Gran Canaria.

Calificación: Apto cum laude.

Resumen: La triangulación de Delaunay en dimensión dos y tres es uno de los métodos de generación de mallas no estructuradas que más se utiliza en la actualidad. Sus características geométricas y algorítmicas la hacen muy adecuada para la aplicación del método de los elementos finitos. El objetivo principal de esta tesis es construir un programa, basado en la triangulación de Delaunay, para generación automática de mallas tridimensionales, y que resuelva de forma eficaz algunos de los problemas intrínsecos al método de triangulación.

El primer capítulo de esta Tesis tiene un carácter introductorio. En él se exponen los conceptos fundamentales y propiedades de la triangulación de Delaunay y sus conexiones con el diagrama de Voronoi.

Entre los numerosos algoritmos existentes, se ha escogido uno basado en el bien conocido algoritmo de Watson. Este algoritmo es de tipo incremental, esto es, crea la triangulación de Delaunay por adición de puntos uno a uno. La ventaja que ofrecen los algoritmos de tipo incremental estriba en su capacidad de adaptación a un proceso de refinamiento local de la malla. La realización de este algoritmo, cuyas principales ideas son presentadas en el primer capítulo, causa problemas cuando los puntos no están situados en "posición general". Estos problemas son aún mayores en la práctica debido a los errores de truncamiento o redondeo que comete el ordenador cuando trabaja con números en coma flotante.

El segundo capítulo se centra en los detalles y las modificaciones del algoritmo expuesto en la introducción. Se muestra la estructura de datos que soporta la información de la triangulación y la forma en que éstos son almacenados y renovados a medida que se van añadiendo puntos. Una parte fundamental de

este algoritmo hace mención a las modificaciones llevadas a cabo para evitar los errores de redondeo mencionados anteriormente.

La definición de dominios poliédricos y la generación automática de puntos son las materias de las que trata el tercer capítulo. La calidad de la triangulación de Delaunay depende de las posiciones relativas que ocupen los puntos; de ahí la importancia que tiene el que éstos se generen de forma adecuada. Otro aspecto que se tiene en cuenta en este capítulo es la conformidad entre la triangulación y la frontera del dominio. Si bien no se dan soluciones definitivas, se apunta un procedimiento basado en la adición de nuevos puntos a la triangulación en las intersecciones entre ésta y el dominio. En los casos tratados se han obtenido unos resultados prometedores aunque subsisten algunos problemas.

El capítulo cuarto está dedicado a la presentación de aplicaciones del mal-lador a diversos ejemplos. Las aplicaciones están acompañadas de resultados numéricos experimentales como tiempo de CPU, medidas de la calidad de la malla, etc.

En los dos últimos capítulos se exponen las conclusiones, algunas líneas de investigación que quedan abiertas y se señalan perspectivas para trabajos futuros.

PRECONDICIONADORES PARALELOS PARA EL
MÉTODO DEL GRADIENTE CONJUGADO PRECONDICIONADO

Doctorando: Cristina Corral Ortega.

Director: Rafael Bru y José Mas.

Defensa: 31 de Mayo de 1995, Universidad Politécnica de Valencia.

Calificación: Apto cum laude.

Resumen: En la memoria se desarrollan dos tipos de preconditionadores paralelos para el método del Gradiente Conjugado Precondicionado para la resolución de sistemas de ecuaciones lineales. El primero de ellos consiste en considerar un número arbitrario de particiones y, a partir de ellas, construir un preconditionador polinomial aditivo, basándonos en la técnica expuesta por Adams y Ong en 1988. Se demuestra que bajo ciertas condiciones sobre las particiones elegidas, el preconditionador polinomial aditivo es una matriz simétrica y definida positiva.

Se han implementado estos preconditionadores sobre un multiprocesador con memoria distribuida Parsys SN1040, considerando dos tipos de particiones distintas, para la resolución de la ecuación de Laplace con condiciones de contorno en el cuadrado unidad, utilizando topología de anillo bidireccional y de malla, llegando incluso, en algún caso, a dividir por dos el número de iteraciones que se obtienen mediante el método de Gradiente Conjugado.

En cuanto al segundo tipo de preconditionadores, estos están basados en una multipartición obtenida mediante Factorizaciones Incompletas de Choleski de la matriz de coeficientes del sistema. Se dan condiciones de convergencia en el caso en que la matriz de coeficientes es una M -matriz. Estos preconditionadores también han sido implementados en el multiprocesador Parsys SN1040, para la resolución de la ecuación de Laplace, considerando tres niveles distintos de llenado (a la hora de realizar la Factorización Incompleta de Choleski), obteniéndose mejores resultados cuanto menor es el número de diagonales que permitimos que se llenen. En todos los casos, el número de pasos del preconditionador óptimo resulta ser 2.

Hemos comprobado que los dos tipos de preconditionadores reducen significativamente el número de condición de la matriz del sistema.

EDPs NO LINEALES ORIGINADAS EN PLASMAS DE FUSIÓN
Y FILTRACIÓN EN MEDIOS POROSOS

Doctorando: Juan Francisco Padial Molina.

Director: Jesús Ildefonso Díaz Díaz.

Defensa: 12 de julio de 1995, Universidad Complutense de Madrid.

Calificación: Apto cum Laude.

Resumen: Se abordan dos problemas de ecuaciones en derivadas parciales no lineales originados en la teoría de plasmas de fusión termonuclear controlada mediante confinamiento magnético en máquinas Stellarator y en la teoría de la filtración en medios porosos. En ambos casos se requieren técnicas propias de EDPs no lineales y resultados de teoría de la medida tales como los relacionados con el reordenamiento decreciente y relativo y con la teoría de funciones de variación acotada que se introducen brevemente.

El primero de los problemas (el concerniente a fusión nuclear) consiste en

encontrar u (escalar) y F (escalar de variable real), tales que

$$\begin{cases} -\Delta u = aF(u) + F(u)F'(u) + bp'(u) & \text{en } \Omega \subset \mathbb{R}^2 \\ u - \gamma \in H_0^1(\Omega) & (\gamma < 0) \\ \int_{\{u>t\}} [F(u)F'(u) + bp'(u)]dx = j(t, \|u_+\|_{L^\infty(\Omega)}) & \text{c.p.t. } t \in [\inf_{\Omega} u, \sup_{\Omega} u] \end{cases}$$

Se obtiene la existencia de soluciones mediante un método de tipo Galerkin aplicado a un nuevo problema de carácter no local que involucra el reordenamiento decreciente de u y relativo de b respecto de u .

En segundo lugar se presenta un estudio analítico del problema de Cauchy-Dirichlet asociado a la ecuación parabólica cuasilínea

$$\frac{\partial b(u)}{\partial t} - \operatorname{div}\{|\nabla u - k(b(u))e|^{p-2}(\nabla u - k(b(u))e)\} + g(x, u) = f(t, x)$$

modelo que aparece en filtración en medios porosos. El resultado principal concierne a la comparación (y por tanto unicidad) de soluciones tales que la derivada temporal de $b(u)$ es una medida de Radon acotada (i.e. $b(u) \in BV_t(Q)$). Finalmente mediante un proceso de regularización parabólica se obtiene la existencia de dichas soluciones.

ALGUNAS CONTRIBUCIONES AL ESTUDIO DEL ERROR
EN LOS MÉTODOS ESPECTRALES: OPTIMALIDAD DE LOS MÉTODOS DE JACOBI
Y ESTUDIO DEL MÉTODO DE "PATCHING"

Doctorando: Generosa Fernández Manín.

Director: Rafael Muñoz Sola.

Defensa: 18 de julio de 1995, Universidad de Santiago de Compostela.

Calificación: Apto cum laude.

Resumen: En esta memoria se pretende aportar algunas contribuciones al estudio de los métodos espectrales. Por una parte generalizamos algunos resultados conocidos en el marco de los espacios de Sobolev sin peso al de los espacios de Sobolev con peso de Jacobi y, por otra parte, en el marco de la descomposición de dominio, analizamos la convergencia del método de "patching" aplicado a un problema modelo. Por último, efectuamos el estudio de la discretización por colocación de un problema mixto Dirichlet-Newmann particular, para lo

cual hemos necesitado obtener unos resultados de aproximación espectral de autofunciones que tienen interés en sí mismos.

En el primer capítulo introducimos las notaciones básicas y algunos resultados que usamos a lo largo de la memoria.

En el segundo capítulo, definimos una nueva familia de espacios con peso y los caracterizamos mediante una norma que depende de los coeficientes de la función considerada en la base de los polinomios de Jacobi. Obtenemos un resultado de aproximación polinomial en estos espacios y un teorema inverso que revela que esa aproximación es óptima. Finalizamos el capítulo caracterizando los espacios en términos de una norma intrínseca, para lo cual utilizamos resultados finos de interpolación de espacios funcionales.

En el tercer capítulo demostramos estimaciones de error para el operador de proyección ortogonal sobre el espacio de polinomios y para la aproximación polinomial de ciertas funciones singulares que surgen al resolver algunos problemas elípticos, y mejoramos los resultados de convergencia para la discretización espectral con el método de Galerkin-Jacobi de un problema modelo. Aunque en la memoria hemos considerado el análisis de un problema simple, los resultados y las técnicas desarrolladas se pueden utilizar para casos más generales.

La segunda parte comienza con el capítulo 4, en el que demostramos una estimación óptima del error cometido al utilizar el método de "patching", con nodos de Gauss-Lobatto-Legendre, en el caso monodimensional. A continuación definimos y analizamos un nuevo esquema, que llamamos pseudo-Galerkin, que nos permite explicitar el error producido por la imposición puntual de las condiciones de acoplamiento.

Un caso bidimensional se estudia en el capítulo 5, donde obtenemos una estimación de error para el método de "patching" aplicado a un dominio cuadrado que se divide en dos rectángulos con un lado común. Para obtener esta estimación, efectuamos un estudio previo de un discretizado del operador de Steklov-Poincaré, el cual nos proporciona también la existencia y unicidad de solución del problema discreto.

En el último capítulo consideramos un problema en un cuadrado con condiciones de contorno mixtas que está directamente relacionado con el método de "patching" y demostramos una estimación óptima del error. La forma de obtener esta estimación nos ha llevado a probar algunos resultados nuevos de aproximación espectral de autofunciones y de ciertas funciones regulares que aparecen al aplicar separación de variables en el problema continuo.

EFFECTOS DE LA ABSORCIÓN EN UNA ECUACIÓN
DE DIFUSIÓN NO LINEAL

Doctorando: Manuela Chaves Tolosa.

Director: Juan Luis Vázquez Suárez.

Defensa: 12 de Septiembre de 1995, Universidad Autónoma de Madrid.

Calificación: Apto cum laude.

Resumen: El objetivo central de la memoria consiste en el estudio del efecto de la absorción en la ecuación de difusión no lineal

$$\Delta u^m - u^p,$$

en el rango de parámetros $1 < p \leq m$.

El comportamiento de las soluciones de estos problemas ofrece una fuerte dependencia del rango de parámetros m y p . En este trabajo se analiza dicho efecto en cuestiones relativas a existencia y unicidad, comportamiento asintótico de las soluciones y de las fronteras libres.

En el capítulo I se considera el problema de Cauchy con datos iniciales no acotados. Se demuestran *condiciones de crecimiento óptimas* para garantizar la *existencia y unicidad* de soluciones. Dichas condiciones son de tipo exponencial simple, $O(e^{\gamma x})$, tanto en el estudio de la existencia como en el de la unicidad, apareciendo un coeficiente crítico γ que es distinto en ambos casos.

El capítulo II se dedica a la construcción y estudio de una solución especial de tipo autosemejante utilizando técnicas de sistemas dinámicos. Dicha solución proporciona un ejemplo de *no unicidad* entre las soluciones no negativas para el problema de Cauchy con dato inicial $u_0 \equiv 0$. Se realiza también, una clasificación de las soluciones autosemejantes; en particular, se descarta la existencia de este tipo de soluciones con soporte compacto.

Por último, en el capítulo III abordamos el estudio del *comportamiento asintótico* cerca de la frontera libre para las soluciones del problema de Cauchy en el rango de parámetros $1 < p < m$. Se demuestra que dicho comportamiento viene descrito por la solución especial construida en el capítulo anterior. Dicha solución también es el modelo autosemejante para el fenómeno de *énfoque en tiempo infinito*.

CONTRIBUCIÓN AL ESTUDIO TEÓRICO
DE ALGUNAS E.D.P. NO LINEALES
RELACIONADAS CON FLUIDOS NO NEWTONIANOS

Doctorando: Rubens Robles Ortega Junior.
Director: Enrique Fernandez-Cara y Francisco Guillén.
Defensa: 13 de septiembre de 1995, Universidad de Sevilla.
Calificación: Apto cum laude.

Resumen: La memoria, que se compone de dos partes, presenta un estudio teórico de algunos sistemas de ecuaciones en derivadas parciales no lineales que describen el comportamiento de ciertas clases de fluidos no Newtonianos.

En la primera parte se considera un modelo para fluidos de Bingham, pseudoplásticos y dilatantes con densidad variable. Para estos modelos, se obtienen resultados de existencia de solución global en tiempo en varios sentidos: solución muy débil, débil y medida valuada. También se estudia la cuestión de la unicidad de solución.

En la segunda parte se considera un modelo para fluidos viscoelásticos de tipo Oldroyd. En cuanto al problema de evolución, se prueba existencia y unicidad de solución fuerte local en tiempo y global para datos pequeños. Para el problema estacionario asociado, se prueba existencia y unicidad de solución fuerte "pequeña" cuando el dato f es suficientemente pequeño. Todos estos resultados son establecidos fuera del marco Hilbertiano. Por último, para el flujo de Poiseuille de un fluido de Oldroyd entre dos cilindros concéntricos, se obtiene existencia y unicidad de solución semi-fuerte (y fuerte) global en tiempo para datos cualesquiera.

RESOLUCIÓN NUMÉRICA DE PROBLEMAS EN DOMINIOS NO ACOTADOS
MEDIANTE ACOPLAMIENTO DE ELEMENTOS FINITOS
Y ELEMENTOS DE CONTORNO

Doctorando: Pablo Pérez Riera.
Director: Javier Valdés y César Omar Menéndez.
Defensa: 3 de noviembre de 1995, Universidad de Oviedo.
Calificación: Apto cum laude.

Resumen: En diversas aplicaciones en Mecánica de Fluidos, Elasticidad, Electromagnetismo, Acústica, etc. se presenta la necesidad de resolver una ecuación en derivadas parciales definida sobre un dominio infinito. Una característica común de estos problemas es que dicho dominio está compuesto por una pequeña región donde el medio puede ser no homogéneo, anisótropo o donde incluso la *edp* puede ser no lineal, y el resto del dominio donde el medio es homogéneo e isótropo y en el que la *edp* es lineal y homogénea. Para proceder a la aproximación numérica mediante técnicas clásicas (diferencias finitas, elementos finitos) es necesario truncar el dominio e imponer unas condiciones de contorno ficticias. En ese caso, el error de aproximación es difícil de estimar, y además el costo computacional es elevado. Una técnica alternativa consiste en acoplar el método de elementos finitos con el de elementos de contorno, dado que este último es adecuado para resolver *edp*'s de coeficientes constantes en dominios infinitos.

En esta memoria se recuerdan los tipos de acoplamiento propuestos por C. Johnson y J.C. Nedelec (1980) y por M. Costabel *et al.* (1991). A partir de ellos, se analiza un método que combina elementos de contorno elementos finitos mixtos con elementos de contorno. Se demuestran resultados de existencia y unicidad, así como estimaciones del error.

Un inconveniente común a las formulaciones globales consiste en que la discretización conduce a sistemas lineales cuyas matrices son *llenas* y de talla muy grande. Una alternativa eficaz consiste en resolver el problema mediante algoritmos iterativos basados en técnicas de descomposición de dominio. Aquí se utilizan dos métodos de este tipo: el método de Schwarz y otro sin solapamiento.

Finalmente, se aplican estas técnicas a la resolución de un problema no lineal de Magnetostática.

Todo aquél que haya realizado/dirigido recientemente tesis doctoral en algún tema de Matemática Aplicada, puede enviar los datos de la tesis y un resumen para su publicación en este boletín. Remitirlos a

sema@posta.unizar.es