

**XIII ADM - XV INGEGRAF**  
*International Conference on*



INGEGRAF

**TOOLS AND METHODS EVOLUTION IN  
ENGINEERING DESIGN**



Cassino, June 3<sup>rd</sup>, 2003  
Napoli, June 4<sup>th</sup> and June 6<sup>th</sup>, 2003  
Salerno, June 5<sup>th</sup>, 2003

---

**E-LEARNING OF COMPUTER-AIDED GEOMETRIC  
DESIGN**

L. Fernández-Jambrina (1), R. López Pulido (2)

(1) Universidad Politécnica de Madrid  
E.T.S.I. Navales, Departamento de Enseñanzas Básicas de la Ingeniería Naval  
lfernandez@etsin.upm.es

(2) Universidad Politécnica de Madrid  
E.T.S.I. Navales, Departamento de Enseñanzas Básicas de la Ingeniería Naval  
debin\_rlpulido@etsin.upm.es

**ABSTRACT**

In this talk we show an example of a thorough e-course on Computer-Aided Geometric Design (CAGD), which makes use of Java animations. The course describes basic algorithms for the layout of Bézier, rational, B-spline curves and surfaces, with special emphasis on ruled, translational, developable, Coons and revolution surfaces.

**Keywords :** CAGD, e-learning, Java, Maple, Cinderella

**1. Introduction**

Computer-Aided Geometric Design (CAGD) is the basis for modern design in most branches of industry, from naval and aeronautic to textile industry. Therefore this subject should be included in most Schools and Faculties of Engineering. In principle, students should just get familiar with some specific design applications, such as Rhino3D, Maxsurf, Foran,... but a thorough knowledge of their capabilities comes from learning at least the algorithms that lie behind the application, even if the student is not to become a developer himself. This is the purpose of CAGD, the geometric description of algorithms for design.

Nowadays industrial design is mainly grounded on Non-Uniform Rational B-Splines (NURBS). These do not require great mathematical developments, just knowledge of polynomial and rational functions and a bit of differential geometry of curves and surfaces. Therefore a course of CAGD could be included in the first semesters of most engineering studies.

One of the major drawbacks of a course on CAGD is that most design applications are too complicated for a basic course, since they usually include more possibilities than can be covered in it. It would be advisable then either to provide simple tools that

implement the basic algorithms of CAGD or to allow the students to write their own codes in a programming language familiar to them (C, Java, Matlab, Maple...).

In this talk we aim to summarize the technical contents of a CAGD course that has been developed by the authors as an optional subject for all sort of students at the Universidad Politécnica de Madrid (U.P.M.). Considering that the U.P.M. campus is not concentrated, but its premises are spread over a vast region of the city and its outskirts, e-learning is surely the best choice for such a heterogeneous group of students and schools. Lessons were not given in any classroom, but the students could follow the course from a website of the university (<http://www.gate.upm.es>), which is specifically devoted to undergraduate and postgraduate courses in the web for students of Spain and the Americas.

## 2. Use of applications

### 2.1 HTML

Since the course is intended for the web, it is written mainly in HTML code, although we have provided some notes for the chapter as PDF files. Obviously raw mathematical stuff is quite involved to be read in a web page and therefore proofs and details have been removed from the HTML files and are included just in the PDF notes, which can be read as an independent book [Fernández 2003].

### 2.2 Maple

In order to generate graphics and animations we have made use of Maple programming language (<http://www.maplesoft.com/>). It may be argued that other applications could suit better our purpose. But we have chosen it for its versatility for producing plots and movies easily in almost every common format (bmp, gif, jpeg, postscript...), for its duality in working with both analytical and numerical data, and, most important, for being the most popular Symbolic Calculus application at Spanish universities, since almost every university has a campus licence for it.

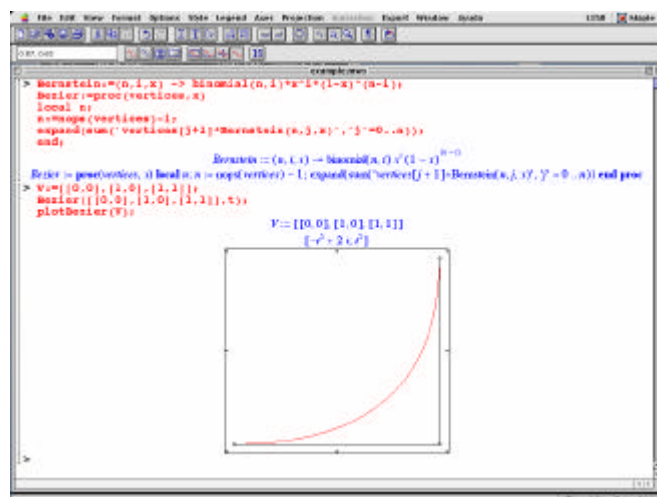


Figure 1. Example of a Maple worksheet

For several months we have built a Maple library of NURBS that comprises the most popular algorithms for curves and surfaces as can be found, for instance, in [Farin 2002]. This library is not fully operational by now and therefore has not been offered to

our students so far, but it serves well for producing examples, exercises, graphics and animations. We expect to complete the Maple library in a few months.

As an example of a simple Maple worksheet we show in Figure 1 the Maple code for a function that calculates the  $i$ -th Bernstein polynomial of degree  $n$  in the variable  $x$ ,

```
Bernstein:=(n,i,x) -> binomial(n,i)*x^i*(1-x)^(n-i);
```

And another function that uses Bernstein for calculating the Bézier parametrization in a variable  $x$  of a polynomial curve of control polygon given by the list *vertices*,

```
Bezier := proc(vertices, x)
  local n ;
  n := nops(vertices)-1;
  expand(sum('vertices[j+1]*Bernstein(n, j, x)',j'=0..n));
end ;
```

These procedures are used to calculate the parametrization of a parabola of control polygon  $[[0,0],[1,0],[1,1]]$  in the variable  $t$ ,  $[-t^2+2t, t^2]$ . Finally, a third procedure is invoked to draw the plot of the parabola and its control polygon.

Graphics can be easily concatenated in order to produce animations, as it is shown in Figure 2, which can be exported as animated GIF files and have been extensively used in the HTML code of the course.

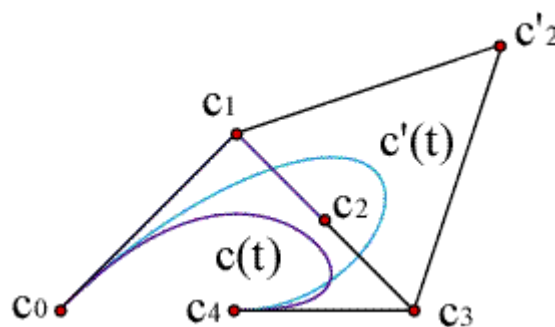


Figure 2. Two shots of an animation

### 2.3 JavaView

Maple graphics and animations are nice, but for an undergraduate course they lack interactivity. For instance, when a surface is drawn, one would request not just to define the best view, the one in which its most remarkable features are shown, but to allow the student to rotate, animate, rescale the surface in order to learn its details, a task that cannot be accomplished with a single view or even with a non-interactive animation.

This inconvenience can be circumvented using the JavaView package for Maple, which allows exportation of Maple graphics as Java applets which can be viewed, animated, rescaled, rotated, translated *ad libitum* by the student (<http://www.javaview.de/>).

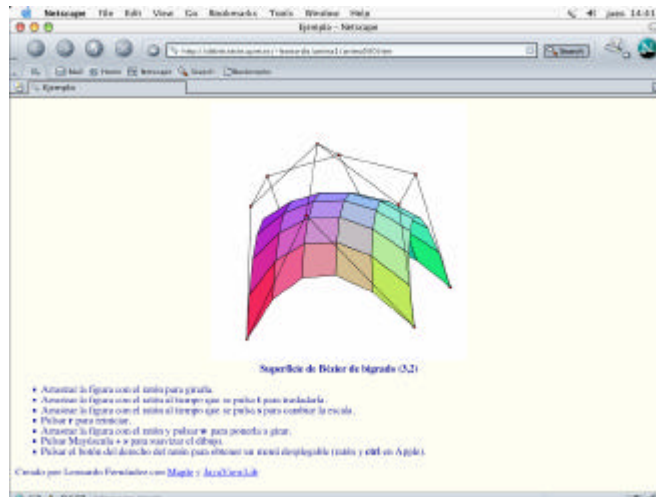


Figure 3. A JavaView graphic

An example of a JavaView graphic is shown in Figure 3.

## 2.4 Cinderella

The possibility of modification of the views of a 3D graphic is interesting, but still the information for a CAGD course would be too static, since the student is not allowed yet to create or modify anything, but to view what has been done before. Since, as it has already been mentioned, we wished to provide simple examples that could reduce the overwhelming amount of information of any design application, we decided to create our own applets in a language that could be implemented universally in web pages.

Java programming language was obviously our first choice, but since we did not want to build new applications from the beginning, we resorted to Cinderella [Richter-Gebert 1999].

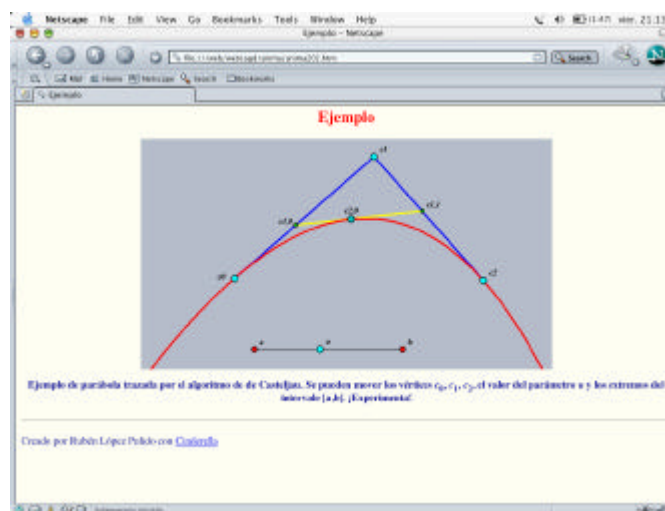


Figure 4. An example with Cinderella

Although it is not a common application out of Faculties of Mathematics, Cinderella offers most of what we might ask. Basically Cinderella works as a Maths teacher would do in front of a blackboard. It allows to draw points, lines, circles, parallel and perpendicular lines, calculation of areas and intersections of lines and curves, animations, ... in order to produce mathematical constructions. And, most important of

all, Cinderella is capable of exporting the results as Java applets that may be inserted in web pages (<http://www.cinderella.de/>).

We have made use of these capabilities to produce several interactive constructions that depict some aspects of CAGD. In Figure 4 you may see a construction made with Cinderella, which draws a quadratic curve from its control polygon with the help of the de Casteljau algorithm. Every point can be modified.

The major drawback that we found in Cinderella was that it cannot deal with analytical expressions, just with synthetic constructions, but it has been surmounted in a recent beta version.

### 3. Contents

The contents of the course are structured in six chapters: one of them is devoted to generalities, three are devoted to curves and two are devoted to surfaces.

#### 3.1 Review of mathematics

The first chapter is a self-contained short review of mathematical concepts that are necessary to follow the main part of the course. Part of the contents of this chapter is surely well known by the students: Cartesian coordinates, ratio of three points, double ratio of four points, whereas other concepts may be new to them: homogeneous coordinates, barycentric coordinates.

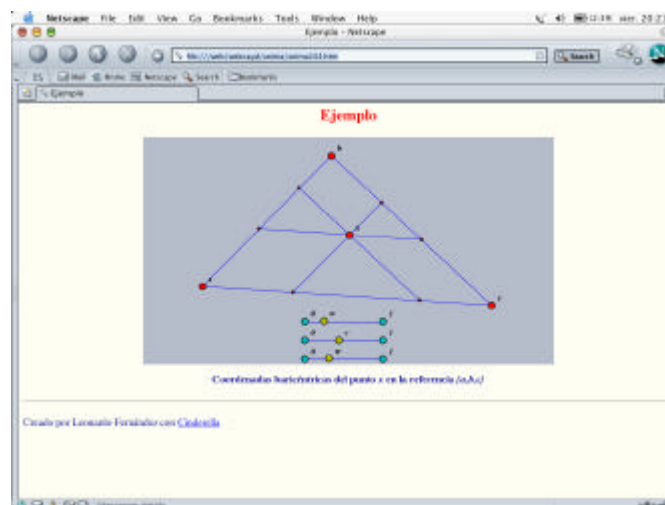


Figure 5. Example of barycentric coordinates with Cinderella

In Figure 5 we see an example of calculation of the barycentric coordinates of a point in the plane done with Cinderella.

#### 3.2 Bézier curves

The second chapter introduces actual CAGD with polynomial curves in terms of Bernstein polynomials and control polygons. The main topics of Bézier curves are displayed as it can be found in every CAGD textbook [Farin 2002], [Hoscheck 1989], [Salomon 1999]. The de Casteljau algorithm is described and the blossom of a Bézier curve is introduced to obtain control polygons for curves. How to link curves polynomial curves is described and applications to interpolation and approximation problems are discussed.

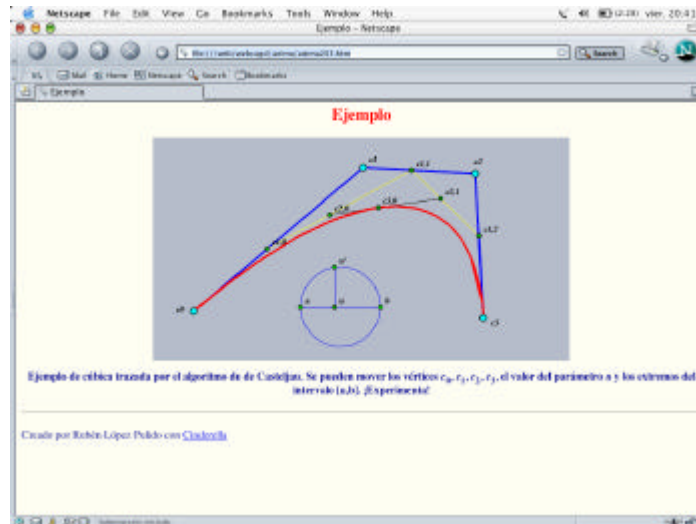


Figure 6. Drawing a Bézier cubic with Cinderella

In Figure 6 we show an example done with Cinderella of the de Casteljau algorithm. With this applet the user may define the control polygon and trace the curve at each value of the parameter.

### 3.3 Rational curves

As a first step of involvement weights are added to Bézier curves in the third chapter in order to construct rational curves. The main features and problems of rational curves are briefly discussed, such as points at infinity, degree elevation and derivatives. A separate section is devoted to conics as rational curves in order to provide a meaning to weights.

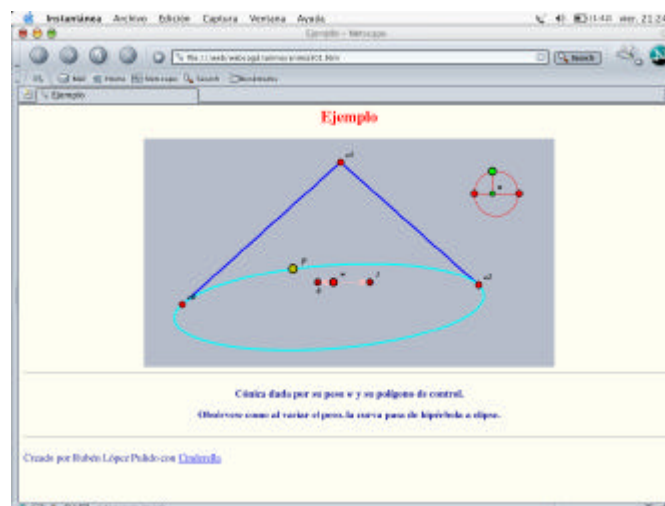


Figure 7. Drawing a conic with Cinderella

In Figure 7 you may see a construction made with Cinderella, which draws a rational quadratic curve from its control polygon and weight. The weight may be raised or lowered to show the curve changing from an ellipse to a hyperbola.

### 3.4 Spline curves

The final topic about curves is piecewise polynomial curves. After reviewing several examples of construction of spline curves, the procedure is generalized via the de Boor algorithm in terms of B-spline polygons and knots. The new features introduced by

spline curves are explored and the algorithm for insertion of new knots is explained and applied to conversion of B-spline polygons to control polygons. A quick reference is made to rational B-spline curves. Finally, B-spline functions are defined and their features are detailed.

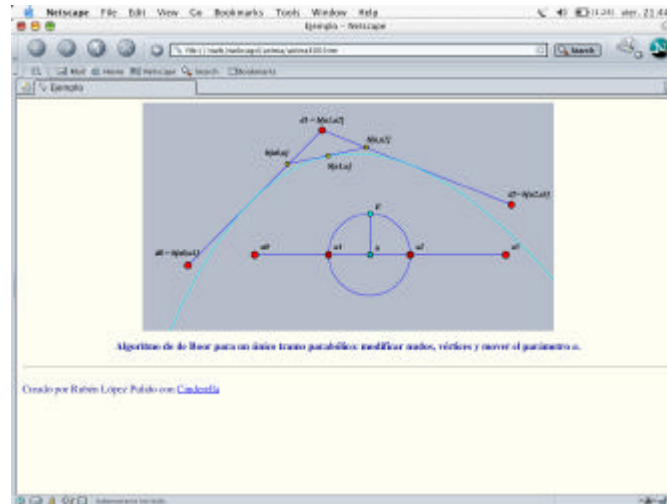


Figure 8. A segment of a spline curve drawn with Cinderella

In Figure 8 we may see the procedure of construction of a segment of a B-spline curve. The user may choose the points of the B-spline polygon and the list of knots.

### 3.5 Bézier surfaces

Polynomial surfaces are introduced in a similar way to Bézier curves thanks to the concept of control nets. Generalization to rational and spline surfaces is nearly straightforward. Topics that were discussed for curves are extended to surfaces: blossoming, calculation of control nets, degree elevation, expressions for derivatives... Interpolation of surfaces is discussed at the end of the chapter. Conditions for joining Bézier patches in terms of their control nets are analyzed.

Only product surfaces are considered in this chapter, since Bézier and B-spline triangles could be too involved for undergraduate students. The study of Bézier triangles is deferred to another chapter.

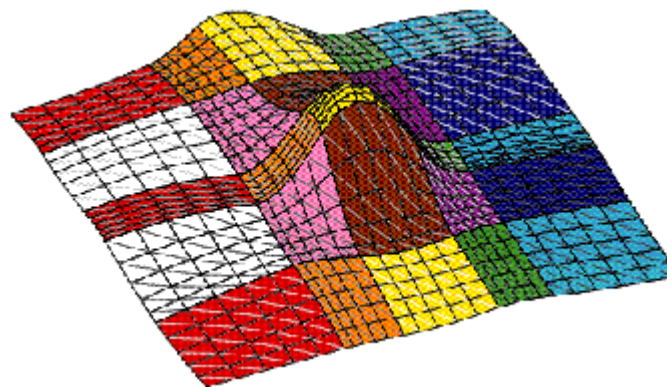


Figure 9. Spline surface

In Figure 9 the local effects of moving a point of the B-spline net of a spline surface are shown.

### 3.6 Generation of surfaces

The last chapter is miscellanea of different ways of generating surfaces. Translational, ruled, developable, Coons surfaces are introduced. An extended section is devoted to surfaces of revolution, both as Bézier and as spline surfaces.

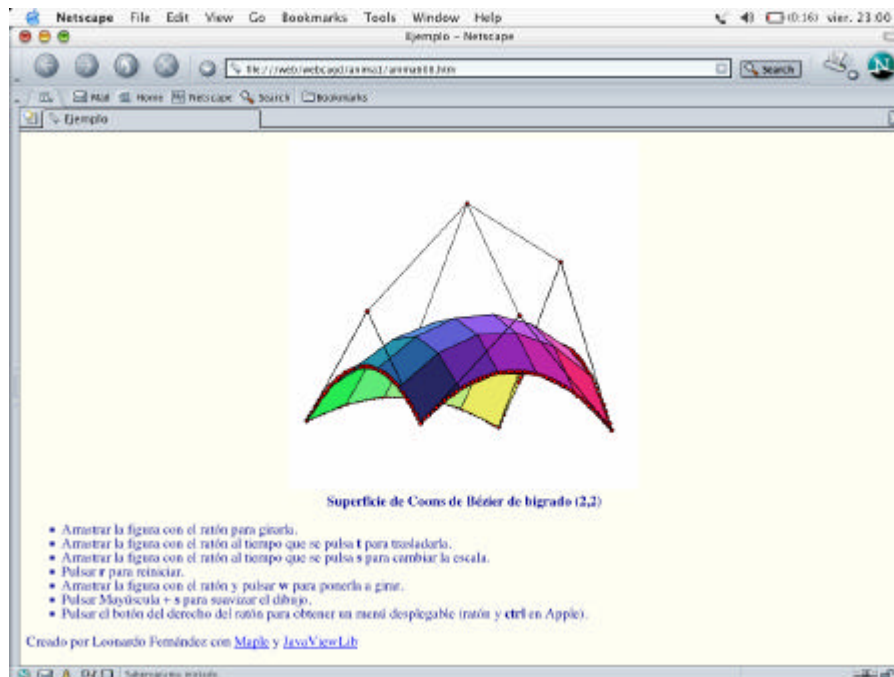


Figure 10. JavaView applet for a Coons surface

In Figure 10 we may see the construction of a Coons surface through four curves done with Maple and JavaView.

## 4. Conclusions and future developments

In this talk we have shown how different applications may be combined to provide an efficient e-course on CAGD. The course has been successfully included this year in the Universidad Politécnica de Madrid studies as an optional subject.

As future developments we wish to reinforce the use of Maple by incorporating Maple applets, maplets, and by exploring the possibilities of the recently released version of Maple 8 for web applications. The new version of Cinderella includes analytical calculations and thereby should make easier the task of creating new applets.

We aim to include new chapters in the course that may make it interesting for postgraduate students. We have in mind a separate chapter on developable surfaces, an issue that is of major importance for naval industry and other branches of engineering that depend on steel sheets. We also wish to include a chapter on Bézier triangles as a different way of constructing surfaces.

### Acknowledgement

The authors wish to thank the G.A.T.E. of the Universidad Politécnica de Madrid for supporting the course in their website and technical assistance. Rubén López Pulido wishes to thank the E.T.S.I. Navales and the Departamento de Enseñanzas Básicas de la Ingeniería Naval for their financial support.

All graphics were made with Maple and Cinderella. Details were embellished with Adobe Illustrator.



## References

Farin G., *Curves and Surfaces for CAGD: a Practical Guide*, Morgan Kaufmann Publishers, San Francisco 2002.

Fernández-Jambrina L., *Notas sobre Diseño Geométrico Asistido por Ordenador*, E.T.S.I. Navales, Madrid 2003.

Hoscheck J., Lasser D., *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley Massachussets 1993.

Richter-Gebert J., Kortenkamp U.H., *The Interactive Geometry Software Cinderella*, Springer Verlag, Berlin 1999.

Salomon D., *Computer Graphics and Geometric Modeling*, Springer Verlag, Berlin 1999.