# Student-Generated Software for Differential Geometry

**Thomas Banchoff and student assistants Jeff Achter, Rashid Ahmad, Cassidy Curtis, Curtis Hendrickson, Greg Siegle, and Matthew Stone**

In a previous article [1], we described the progress made by a team of students working together with a professor to develop **Vector,** a software program for third semester calculus, to be used in an interactive laboratory environment. The original program enabled students to display graphs of space curves and functions of two variables, and subsequent improvements have added a number of features for displaying geometric objects associated with curves and surfaces. An enhanced version of the original program continues to be used as a supplementary tool in several courses in third-semester calculus at Brown University, with consultants on hand to assist students using the program and to receive suggestions for improving its effectiveness. At present we are using this program in an interactive laboratory associated with the introductory course in the differential geometry of curves and surfaces.

During the current semester, a new opportunity in the development of the program has arisen in the undergraduate differential geometry course. Two of the students in the project are enrolled in the course and are acting as laboratory assistants for a weekly hour-long session devoted to the specific topics of the course. The two dozen students in the class can return to the fifty-unit laboratory to run the program at other times when the laboratory is available to all students on a first-come first-serve basis. Students can work individually or together on assignments which require them to investigate the behavior of a curve or a surface or a family of such objects. Several of the challenges we have encountered are interesting in their own right, as we seek the best ways of utilizing the growing capabilities of the machines and the programs. By discussing these topics here, we wish to give some feeling for the way that this project is progressing as a true collaboration between students and instructors, and to show some of the ways our experience with the computer laboratory environment suggests changes in the choice of topics and the presentation of the subject matter.

## The Cardioid Series

In dealing with parametric curves, it is often desirable to investigate not just one object but rather a family of curves. A particularly interesting example is the family which includes the cardioid. We define a family of polar coordinate function graphs depending on one parameter $c$, by

$$X(t) = ((c + \cos(t))\cos(t), (c + \cos(t))\sin(t)).$$

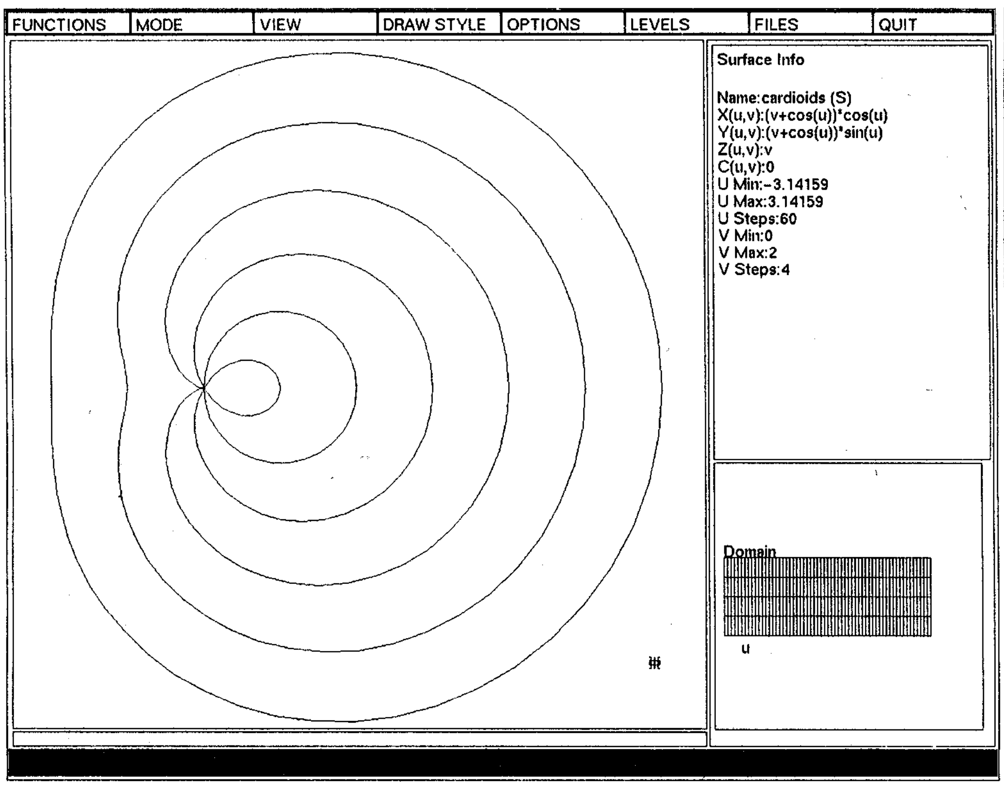
The value $c = 1$ (or $c = -1$) gives a cardioid with a cusp at the origin. What is the behavior of the other curves in the family?

*Tom Banchoff has been a professor in mathematics at Brown University since 1967, specializing in the geometry of surfaces in 3- and 4-dimensional space.*

*The student assistants are mathematics-computer science majors at Brown University. Rashid Ahmad graduated in 1988 and is a student at the Columbia University School of Medicine. Greg Siegle is a junior and Jeff Achter, Cassidy Curtis, Curtis Hendrickson, and Matthew Stone are sophomores.*

Using **Vector**, students can enter the coordinate functions for the above equation, then choose various values of $c$ and display the corresponding curves, one at a time or several at once. An example is shown in Figure 1. One of the primary aims of the course is the analysis of singularity behavior, so it is especially important to analyze the curves near the critical position, say at $c = .9$ and $c = 1.1$. In the first case, the curve is locally convex with a double point at the origin; in the second, the curve is one-to-one, but it has a pair of inflection points and an interval where the curvature is negative. This behavior is typical for deformations of cusps, and students will recognize it again and again during the course.

**Figure 1**




Are there any other interesting choices for $c$? A student entering various other values of $c$ will easily note that there is a symmetry, and curves corresponding to opposite parameter values are congruent. This suggests a special treatment of the value $c = 0$, which gives a doubly-covered circle, unfolding into a curve with one double point when $c$ is a small positive or a small negative number. What about large values of $c$? Will there always be inflection points if $c$ is greater than 1? Computer investigation indicates that this is not so, and it seems that the curves are convex for all values of $c$ greater than 2.

Naturally in a mathematics course it is not enough only to observe these phenomena; we also have to prove that what we observe is true. Experimentation with polar coordinate function graphs suggests a criterion for double points, either $r(t+\pi) = -r(t)$ for some $t$ or $r(t) = 0$ for two different values of $t$. These conditions show that there will be double points when $|c| < 1$, as observed.

The existence of inflection points is equivalent to the vanishing of the numerator of the curvature

$$\begin{aligned} y''x' - x''y' &= -rr'' + 2r'^2 + r^2 \\ &= -(c + \cos(t))(-\cos(t)) + \sin^2(t) + (c + \cos(t))^2 \\ &= 1 + c^2 + 3c\cos(t) + \cos^2(t) = 0. \end{aligned}$$

This will have solutions exactly when $1 \le |c| \le 2$, as predicted by the images on the computer screen.

## ***Animating Parameter Changes***

In addition to keying in desired values of a parameter, a student can set up a sequence of examples by instructing the computer to change $c$ from a beginning value to an end value in a certain number of steps. The images can then be played back to give an animated view of the deformation represented by the change of parameter values. This technique enables a student to "unfold" a singular phenomenon which arises naturally in the course of a one-parameter deformation.

## Parallel Curves for Function Graphs

Animation techniques are especially effective when a deformation is related to a physical phenomenon, such as the propagation of wave fronts in the neighborhood of a curve. This phenomenon leads to the concept of *parallel curves*, where the parallel curve at distance $r$ is obtained by moving $r$ units along the unit normal at each point of the curve. Classical treatments of this subject emphasized the fact that if the distance is sufficiently small, the parallel curve to a smooth curve is smooth. With computer graphics, we can deal with a larger range of phenomena, and we can pay much more attention to the important subject of singularities of curves and families of curves.

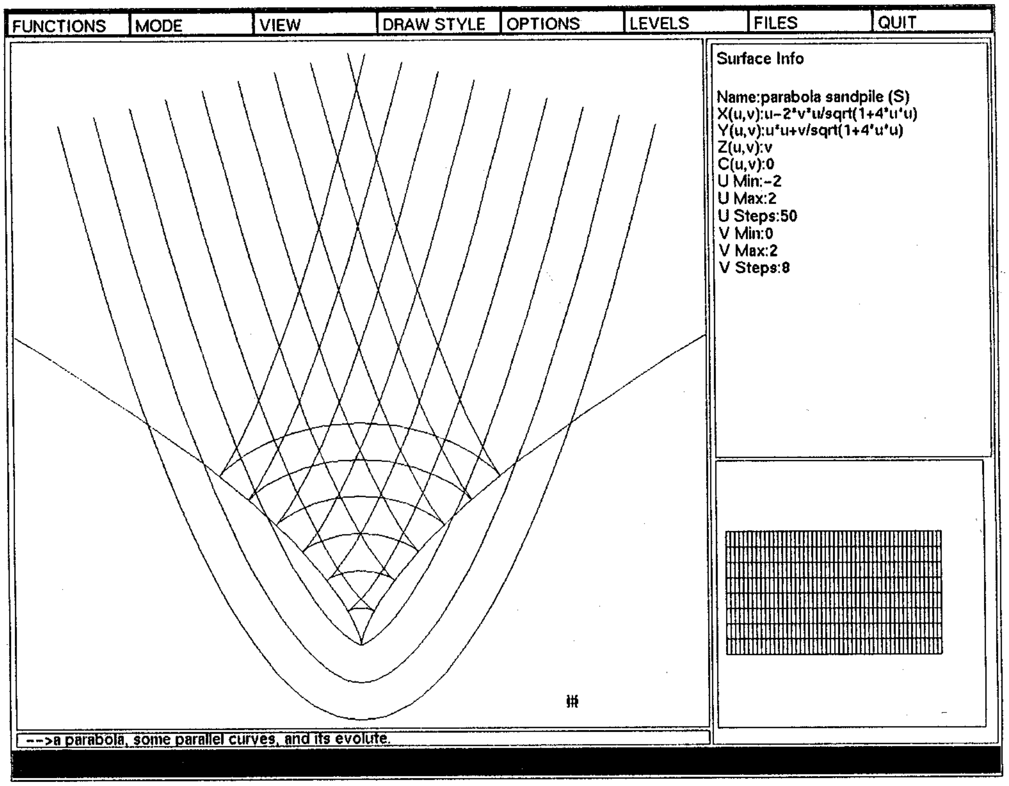
A student can use **Vector** to enter in the equations for the coordinates of a parallel curve, with the parameter $c$ as the distance to the parallel curve. Symbolically we may write $X_c(t) = X(t) + cU(t)$, where $U(t)$ is the rotation of the unit tangent vector $T(t) = X'(t)/|X'(t)|$ by $\pi/2$ radians. In early versions of **Vector**, students entered the explicit equation for each coordinate separately. The current version allows students just to enter the original curve and have the program calculate the auxiliary vectors automatically. By choosing various values of $c$, it is possible to see where the parallel curve develops cusps, and where the parallel curve intersects itself.

For the parabola $X_c(t) = (t, t^2)$, the student would enter the equations $x(t) = t - 2ct/\text{sqrt}(1 + 4t^2)$, $y(t) = t^2 + c/\text{sqrt}(1 + 4t^2)$, $-2 \leq t \leq 2$, and then select various values of $c$. For small values of $c$, the curve appears to be smooth, up to the value $c = 1/2$ where the curve seems to have a corner. Closer inspection shows that the curve is smooth although very flat at the point $X_{1/2}(0)$. For larger values of $c$, the curve has a pair of cusps and a single intersection point, a fact that is easy to check analytically.

In order to study the relationship between a curve and its parallel curve, it is important to display both of them at the same time and to relate them to the same coordinate system. When first given a curve, **Vector** computes the maximum and minimum $x$- and $y$-coordinates and displays the image on the largest possible square screen. It is possible to choose an option "View in Same Space" so that subsequent curves are referred to the coordinate system of the first curve, even if they do not fit on the screen. It is also possible to change the roles of curves so that the screen size is determined by any one of them. We can also resize the screen manually by selecting a square in which the visible screen will be redrawn, or zoom in on a particular section by selecting a square which is then expanded to fill the screen. At any stage in the investigation, it is possible to display the coordinate axes, or to choose a viewing space centered at the origin. The "bird's eye view" option shows the projection into the $x$-$y$-plane.

In general the variation of the vector $U(t)$ gives information about the way the curve deviates from a straight line, and we may define the *(geodesic) curvature* $k_g(t)$ by $U'(t) = -k_g(t)X'(t)$. It is then clear that the parallel curve is smooth, with $X'_c(t) = (1 - ck_g(t))X'(t)$, so the tangent lines at corresponding

**Figure 2**




points are the same and the direction of the unit tangent vector to the parallel curve is the same as that of the original if and only if $(1 - ck_g(t))$ is positive. The parallel curve has a singularity if the distance $c = 1/k_g(t)$, the *radius of curvature* of the curve at $X(t)$. The locus of singularities of the parallel curves is called the *focal curve* or *evolute* curve of $X$.

To illustrate this important fact, it is possible to display the curve and its evolute $E(t) = X(t) + (1/k_g(t))U(t)$ simultaneously, along with several parallel curves, as shown in Figure 2, or it is possible to set up an animation to show the cusps of parallel curves moving along the evolute.

## *Parallel Regions for Plane Curves*

Since **Vector** was originally written to display surfaces, it is easy to use the two-parameter capability to create an entire family of parallel curves. We define a surface by $Y(u,v) = X(u) + vU(u)$, where $X(u)$ and $U(u)$ represent the curve and its unit normal vector in the plane, and where $v$ goes from 0 to $c$ in a certain number of steps. We always have the option of showing only the curves $v$ = constant, and this family gives the desired set of parallel curves.

We get a bonus from this representation if we show instead the curves $u$ = constant, along with the original curve. We then have the collection of normal lines to the curve, and the curve or points where nearby normals intersect is quite evident. We can then show that this singularity curve is the evolute of the original curve. Then $Y_u(u,v) = X'(u) + vU'(u) = (1 - vk_g(u))X'(u)$ and $Y_v(u,v) = U(u)$. Since $X'(u)$ and $U(u)$ are always linearly independent for a regular curve $X$, the only singularities occur if $v = 1/k_g(u)$, i.e. at the points of the evolute $E(u) = X(u) + (1/k_g(u))U(u)$.

At this stage, we can get additional information by using the color capabilities of the machine. We may assign colors to points of the curve according to the values of the parameter, and then color the points on the evolute similarly to establish the correspondence. This is especially clear when we use the same colors on the rays going out perpendicular to the curve.

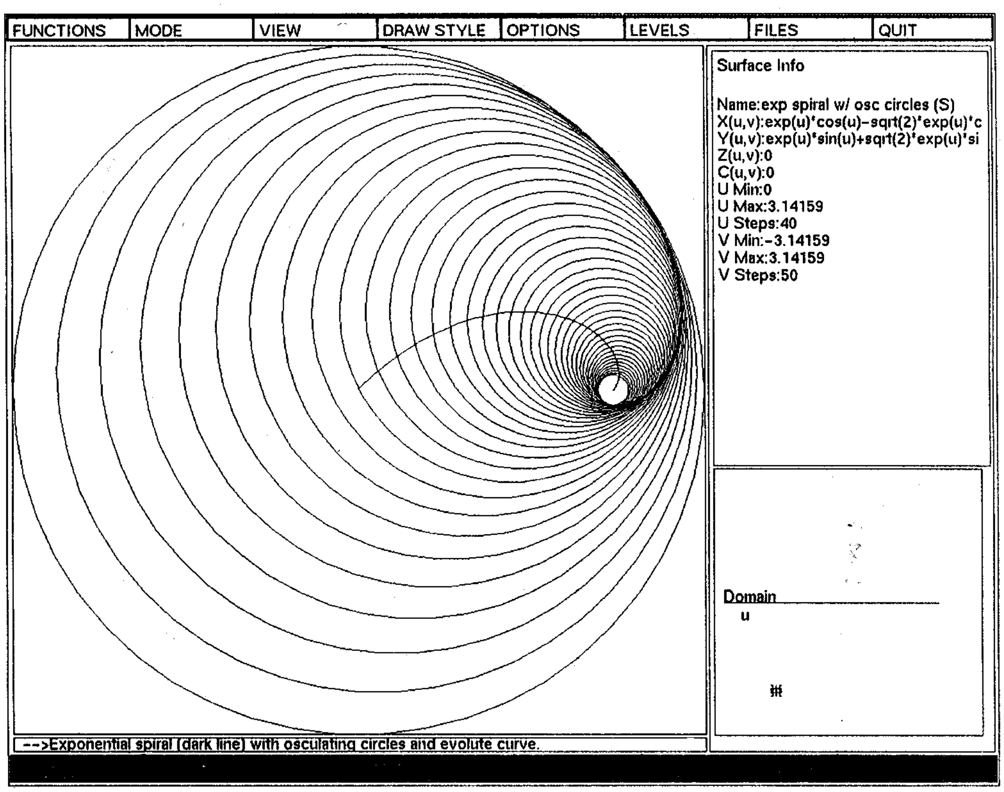
## *Evolute Curves for Epicycloids*

One of the most dramatic "discoveries" that appears from the use of this program is the evolute phenomenon for the family of epicycloids, $X(t) = ((1 + c)\cos(t) - c\cos((1 + c)t/c), (1 + c)\sin(t) - c\sin((1 + c)t/c))$. Entering in the equations of the curve together with its normal rays produced images which are unmistakably similar to the original curve, no matter what the value of $c$ may be. Only if $c$ is integral will the curve close off in the interval $0 \leq t \leq 2\pi$, but for other rational values of $c$, the curve will close an appropriate multiple of $2\pi$, producing some striking images of curves and their (similar) evolutes. Of course it is then possible (for the better students) to prove the theorem that the evolute is indeed similar to the original curve.

## *Nesting of Osculating Circles of a Spiral*

It is a fact that osculating circles of a spiral with monotonically increasing curvature are nested, i.e. the best approximating circle at the beginning of such an arc completely contains the corresponding circle at the endpoint. This result is surprising to many students. If a student draws a curve on a paper or blackboard and sketches in the osculating circles at two nearby points, the circles almost always appear to cross, even though the theorem predicts that they wil not intersect. **Vector** makes it possible to illustrate the theorem and to see what the collection of osculating circles really looks like. A student can enter a spiral arc, say the positive half of a parabola $X(u) = (u, u^2)$, $0 \leq u \leq 2$, along with a pair of its osculating circles, say the one at the origin, with radius 1/2 and center (0, 1/2) and the one at (1, 1) with center at (-4, 7/2) and radius $2/5^{3/2}$. The smaller circle is contained in the larger, and we don't need a computer to show that. But the computer can show even more. The program is set up to show surfaces, so it is possible to build an entire set of curves and to show as many of them as we wish. We can show not just two circles but the family of all osculating circles at all points of the parabola, and the visual evidence that they do not meet is compelling, as shown in Figure 3.

**Figure 3**




There are various capabilities of the machine that make these phenomena even clearer. We can color the points of the original curve according to the parameter value, then color the osculating circle at a particular point with the same color. We can also show a sequence of bands between successive osculating circles to make the nesting property even more evident.

## *Further Topics in the Geometry of Curves and Surfaces*

Up to now we have discussed only the two-dimensional capabilities of the machine. We can gain even more insights by looking at objects in three-space. We first study surfaces associated with curves, such as the *tangential surface* $X(u) + vT(u)$, where $T(u)$ is the unit tangent vector of a space curve, or the *normal surface* $X(u) + vP(u)$, where $P(u)$ is the principal normal. We can then work with tube surfaces, like the *normal tube*

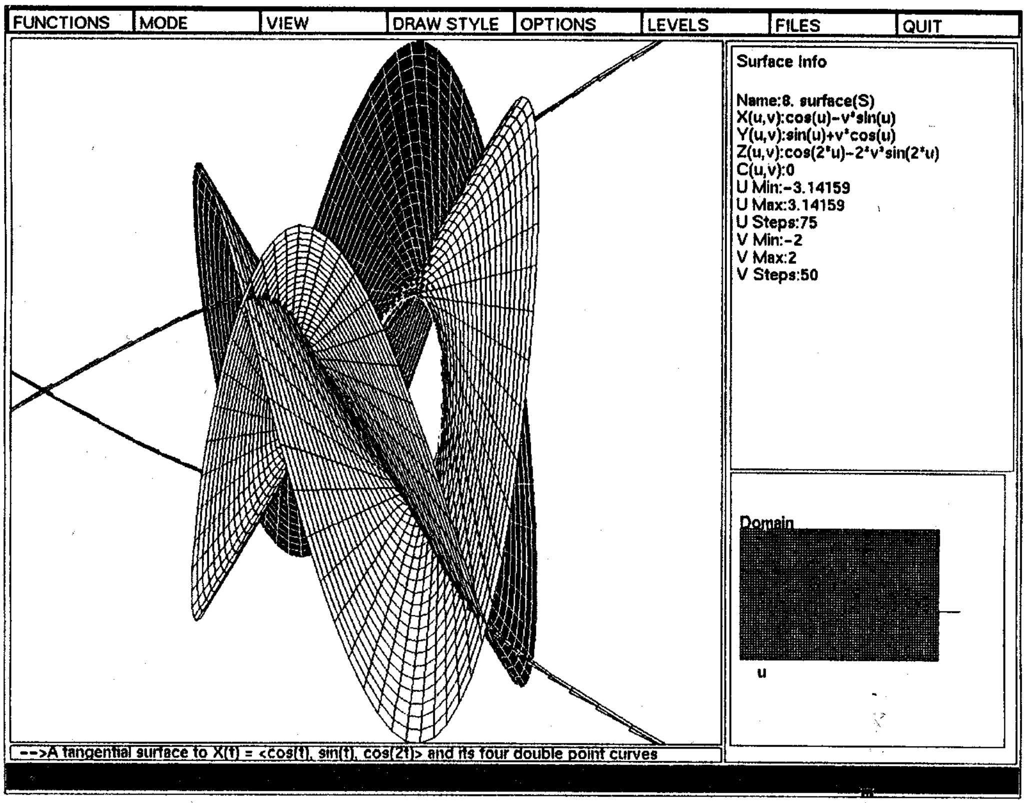
$$X(u) + r\cos(v)P(u) + r\sin(v)B(u)$$

which are the analogues of the parallel curves in the plane, in order to study the curvature and torsion properties of curves. We can define more general tubes, such as the *curvature tube*

$$X(u) + (1/k(u))(\cos(v)P(u) + \sin(v)B(u)),$$

a useful technique for modelling the growth of shells. A tangent surface is shown in Figure 4, and a normal tube is shown in Color Plate 3.

**Figure 4**




We can then go on to study surfaces in their own right, including parallel surfaces and focal surfaces of function graphs and of parametric surfaces. All of these capabilities are already available on **Vector** once we enter the appropriate combinations of functions. At the same time, it is clear that these more complicated topics are stretching the capabilities of the program, and we can anticipate a further redesign in the future. We look forward to the ability to work with vector functions, and to calculate quantities like curvatures and principal directions, without having to enter the explicit equations for each coordinate. (This capability existed to a certain extent in the program EDGE developed for an earlier version of the interactive laboratory environment in differential geometry, as described in [2].)

We have learned a great deal in the development of these programs, using the ideas of differential geometry to suggest new directions for the project in the calculus of curves and surfaces. As mentioned in previous articles, the interaction between faculty and students in this collaboration is one of the most rewarding aspect of the whole enterprise. We look forward to the next phases of our project, and the continued use of interactive software in geometric investigations.

## References

[1] Banchoff, Thomas, and Student Associates, “Student-Generated Interactive Software for Calculus of Surfaces in a Workstation Laboratory,” *UME Trends*, (August 1989), 7-8.

[2] Banchoff, Thomas and Richard Schwartz, “EDGE: The Educational Differential Geometry Environment,” in *Educational Computing in Mathematics*, T. F. Banchoff, et.al., (eds), Elsevier Science Publishers B.V., North Holland, 1988, 11-30.