A User Guide to the Curve Subsystem of DesignMentor

Dr. C.-K. Shene
Assistant Professor
Department of Computer Science
Michigan Technological University

Program Designed and Implemented by Yuan Zhao

© 1997 Yuan Zhao and C.-K. Shene

Click here to download a copy of DesignMentor

Select the topics you wish to review:

Fundamentals
- A Quick Tour
- The Drawing Canvas
- Creating Curves
- Working with Multiple Curves
- Saving and Loading Your Work
- Clamped, Open and Closed Curves
- Tracing the Curve - de Casteljau’s and de Boor’s Algorithms

Shape Modification
- Modifying Control Points
- Modifying Weights
- Modifying Knots

Moving into Space
- Various Useful Display Options
  - The First and Second Derivative Curves
  - The Moving Triad
  - The Curvature Sphere
  - The Roller Coaster Effect

Changing the Color Scheme

Advanced Features
Knot Insertion
Degree Elevation
Curve Subdivision

Windows
The Display & Tracing Options Window
The Partition of Unity Window
The Triangular Computing Scheme Window
The Hodograph (First Derivative) Window and Second Derivative Window
The Change Color Scheme Window
The Profile Instances Control Window

Menu Structure
File
Curve
Edit
Techniques
Degree
Window
Help

Please send comments and suggestions to shene@mtu.edu

A Quick Tour

This program uses the left mouse button exclusively

After launching the curve system you should see the following two windows:

The left one is the drawing canvas on which you design your curve. The right one is the Display & Options Window, which contains various display options.

Exiting the Program

To exit this program, select File followed by Exit.

Creating Control Points

The drawing canvas is the $xy$-plane with the $x$-axis running from left to right and the $y$-axis running from bottom up. The coordinate origin is at the center of the drawing canvas.

The first step of designing a curve is putting some control points on the drawing canvas, the big square area. This requires the following steps:

1. Select Edit followed by Create. The pointer changes to a cross indicating that the program is in the insert mode.
2. Move the pointer to the drawing canvas and click at various locations. Each click creates a control point shown as a little square with a number attached, the control point number. Note that control point numbers start with 0.
3. After adding all control points, the curve system must be told about the curve type you want. Although this program supports Bézier, rational Bézier, B-spline and NURBS curves, Bézier curves will be used in this tour exclusively. By default, the curve type is Bézier. But you can always change this default by selecting Curve, followed by New Curve Segment, followed by the type of the curve (e.g., Bézier, rational Bézier, B-spline and NURBS), followed by Show Curve Segment. The curve defined by the control points will be shown. In the following, six control points and the corresponding Bézier curve are shown.
The degree of the Bézier curve is shown in the **Degree** button. Since there are six control points, the degree of this Bézier curve is 5.

### Changing the Positions of Control Points

Changing the position of a control point is easy, even though a curve has already been displayed. To move a control point, the first thing to do is leaving the insert mode. To do so, choose **Edit** followed by **Move**. After this, the pointer is changed to a north-west pointing arrow indicating that the program is in **move** mode.

In move mode, one can move the pointer to a control point, drag it to a new location, and then release the mouse button. As the control point moves, the shape of the curve changes on-the-fly. Let us move control point 3 downward, the shape of the curve is pulled downward as shown below:
Control Polyline and Convex Hull

The **Display & Options Window** has a number of buttons and is subdivided into four parts. Selecting one of these buttons by clicking on it either activates or deactivates the corresponding option. If an option is active currently, the button shows a $\times$. Clicking on an active (resp., inactive) button will deactivate (resp., activate) it.

Activating **Convex Hull** displays the convex hull of the control points. Activating **Control Polygon** displays the control polyline. The following shows the control polyline and convex hull of the given control points:

![Control Polyline and Convex Hull](image)

**Curve Tracing**

To trace the curve use the vertical slider. The range of this slider is 0 and 1. The current position of $u$ is marked by a small triangle, the $u$-**indicator**, and its value is shown in the middle of the slider. As the $u$-indicator moves along the slider up and down, its value is shown and a small dot, which indicates the point on the curve corresponding to the current value of $u$, moves on the curve. In the following figure, the $u$-indicator is moved to 0.45 (i.e., $u = 0.45$) and the corresponding point on the curve is near control point 2.
De Casteljau's Algorithm

If de Casteljau's alg. is activated, all intermediate polylines of the computation of de Casteljau's algorithm will be shown. As \( u \) moves, this net of polylines moves as well. The following shows a snapshot of this feature. Note that different polylines are shown in different colors.

To make the polylines clearly shown, one might activate w/ point to show the vertices of polylines.
Partition of Unity

Since Bézier basis functions play an important role in generating the curve, this program can display these basis functions and partition of unity. To do so, choose **Window** followed by **Partition of Unity**. The **Partition of Unity Window** appears and shows all basis functions. To dispose this window, select the **Dismiss** button at the lower left corner of this window.

In our example, there are six control points and hence six basis functions, one for each control point. Each of these six basis functions are shown with a different color. As $u$ moves by dragging the $u$-indicator with the vertical slider, its position will also be shown in this partition of unity window with a vertical bar. At the right end of this window, values of all basis functions at $u$ are arranged into a vertical line segment showing the way of partitioning of unity. As $u$ changes, the way of partitioning of unity changes as well. The above figure shows the way of partitioning of unity at $u = 0.5$. 
The Drawing Canvas

This curve program provides you with a drawing canvas and several control mechanisms for you to design and manipulate curves. The drawing canvas is the big square area as shown below.

The menu bar consists of six items File, Curve, Edit, Techniques, Degree and Window. There is a Help button in the upper right corner for showing other important information concerning this program.

The vertical slider has two purposes. First, the right side is for tracing the curve. When the small triangle, the \textit{u}-indicator, is moved, the value of \( u \) is shown and the corresponding point \( p(u) \) will trace the curve. You can either drag the \( u \)-indicator for fast move or click the Up and Dn (for down) buttons for step-by-step move. The values correspond to Up and Dn are 1 and 0, respectively. The left side of this slider will show the knot vector if the curve being displayed is a B-spline or a NURBS curve.

The bottom part has two sliders and a number of buttons. Buttons X, Y, Z and W display the coordinates and the weight of the selected control point. They also indicate the reference axis with respect to which a translation or a rotation will be performed. Coupled with the two horizontal sliders, buttons Point, Curve and Scene permit transformations to be applied to a selected control point, a curve and the complete scene, respectively.

A detailed discussion and the use of these buttons and sliders will be presented in Moving into Space. If your operations are restricted in the xy-plane, these buttons and sliders are not necessary. However, they are required elements for cross section design, since this type of design techniques are based on curves in space.
Creating Curves

To create a curve in the xy-plane, you need the following steps:

1. Select **Edit**, followed by **Create**. This will put the curve system in the **insert** and the cursor is changed to a cross. Note that the menu item shows **Edit: Create**, meaning that the system is in the insert mode.

2. The second step is to specify a curve type. This system supports four types of curves: Bézier, rational Bézier, B-spline and NURBS curves. All four types require a set of control points.

   Select **Curve**, followed by **New Curve Segment**, followed by the curve type you want. For example, if you choose the curve type to be **B-Spline**, then the menu item shows **Curve: B-Spline**.

3. For Bézier and rational Bézier curves, the degree of the curve is the number of control points minus 1. For example, if a Bézier curve is defined by nine control points, its degree is 8. For B-spline and NURBS curves, one need to specify the degree of the desired curve. The default degree for a B-spline and NURBS curve is 3. But, you can change this value by selecting **Degree**, followed by the desired degree. This system supports degree up to 10 for B-spline and NURBS curves, which is sufficient for more design purposes. If we choose 4, the menu item will show **Degree: 4**

4. Now, you can move the cross pointer into the drawing canvas and click to put control points there. In the figure below, we put 13 control points on the drawing canvas. Please note that the control point IDs start with 0. As a result, the control points are numbered from 0 to 12.

5. Finally, we can display the curve by selecting **Curve**, followed by **Show Curve Segment**. We have three choices: **With Uniform Knots**, **With Clamped Knots**, and **With Closed Knots**. Meanwhile, let us select **With Clamped Knots**, since our study centers around this type of curves. Then, we shall see the following B-spline curve of degree 4, with clamped knots, on the drawing canvas.

6. Recall that the right hand side of the vertical slider is for curve tracing and the left hand side shows the knots. Each knot is shown in a form of m (d,ddd), where m and d,ddd are the multiplicity and value of a knot, respectively. After a curve is generated, all knots are equally spaced.

As mentioned earlier, Bézier and rational Bézier curves do not require a degree. One can just put control points on the drawing canvas. As you are clicking, the curve will be generated on-the-fly. So, creating Bézier and rational Bézier curves are simpler than creating B-spline and NURBS curves. The following figure shows a Bézier curve generated from the same set of control points that were used to created the above B-spline curve. Note that since there are 13 control points, this Bézier curve should have degree 12 as shown in the **Degree: 12**.
If you compare this Bézier curve with the above B-spline curve, you will see that the B-spline curve of degree 4 follows the control polygon more closely than the Bézier curve of degree 12 does. This is a good reason for using B-spline and NURBS curves, because it allows to use a lower degree curve to follow the given control polyline closely. However, Bézier curves are simpler and easier to understand and implement.
Working with Multiple Curves

This system allows you to design and work on a number of curves at the same time on the drawing canvas.

Design the first curve has no mystery. Just follow the procedure described in Creating Curves. To design the second and all subsequent curves, use the same procedure:

1. Select Curve, followed by New Curve Segment, followed by the type of curve you want.
2. Select a degree (if it is a B-spline or NURBS curve) and put a number of control points on the drawing canvas.
3. Select Curve, followed by Show Curve Segment, followed by the type of knots (i.e., clamped, open or closed) you want. Then, a second curve will appear on the canvas.

Therefore, creating the second and all subsequent curves is essentially the same as creating the first one.

The left figure below is a Bézier curve of degree 6. Then, follow the above procedure to design a B-spline curve of degree 4. This is shown in the right figure.

Switching Among Curves

Now we have more than one curves on the drawing canvas. We really need some way to select and work on one of them. This comes the concept of the current curve. The current curve is the curve you can edit and transform. If there are more than one curves on the drawing canvas, the current curve is always thicker with brighter color. When you create a new curve, that curve is the current curve. For example, in the right figure above, when the B-spline curve of degree 4 is created, it becomes the current curve.

Since only the information about the current curve (i.e., type, degree and knot vector) are shown in various buttons and sliders, you can work on the current curve only.

In the system, all curves are stored in the order they are created. Thus, given any curve, one can always find its next and previous ones. The next (i.e., previous) curve of the last (resp., first) curve is the first (resp., last) curve on the canvas. There are two ways for changing the current curve. The first one uses the menu items, while the second uses the Display & Tracing Options Window. Select Curve, followed by Next Curve Segment (resp., Previous Curve Segment) to make the next (i.e., previous) curve the current curve.
Or, you can bring up the *Display & Tracing Options Window* by selecting *Window* followed by *Display & Tracing Options*. Near the bottom, you should see two buttons *Next Curve* and *Prev Curve* as shown in the figure above. Clicking on the former moves to the next curve, while clicking the latter moves to the previous curve.
This system supports saving the complete scene to a file and loading it back at a later time. Or, you can load a saved scene and merge it with the scene currently on the drawing canvas. In the former case, the new scene will overwrite the existing one, while in the latter, the new scene will contain all curves currently on the drawing canvas and all curves loaded from a file.

When designing a complex scene that involves many curves (i.e., profile and trajectory curves for cross section design), you may want to save your intermediate design periodically so that you will be able to restart from a previous good design. Or, you can save a number of key curves, one in each scene, and import them back one by one into a complete scene. To help you do this, this system provides you with three options: (1) **Save Scene** saves the current scene, (2) **Load Scene** loads a saved scene back to the drawing canvas, overwriting the existing one, and (3) **Import Scene** brings all curves stored in a file into the existing scene.

### Saving a Scene

To save all curves currently on the canvas, select **File**, followed by **Save Scene**. Then, you will see a dialog box asking you for a file name. If you have previously saved or loaded a file, that file's name will be shown.

Type in a file name and click **Ok**. All curves on the drawing canvas will be saved to the indicated file.

### Loading a Scene

To load a scene back from a file, select **File**, followed by **Load Scene**. You will also see a dialog box asking for the file name to be loaded. Type in a file name and click **Ok**. The scene stored in the file will replace the existing scene on the drawing canvas.

By default, the scene is saved as a text file. However, there is a button **Binary** for saving the scene into a binary file. This is mainly for compatibility purpose and new versions of **DesignMentor** will not save files in binary form.

### Import a Scene

If you want to add the curves in a saved scene into the current one, use import rather than load. To import a scene, select **File**, followed by **Import Scene**. Type in a file name and click **Ok**. The scene stored in the file will be added to the existing scene on the drawing canvas. As a result, the drawing canvas will contain all curves from the current scene and the saved scene.

By default, the file is saved as a text file. However, there is a button **Binary** for importing a binary scene. This is mainly for compatibility purpose and new versions of **DesignMentor** will not save files in binary form.
If the desired curve is a B-spline or a NURBS curve, you have three choices: **With Uniform Knots**, **With Clamped Knots**, and **With Closed Knots**. This indicates the different way of arrange the knots.

If you choose **With Uniform Knots**, the curve will be tangent to the first and the last line segments of the control polygon (the above left figure). Please note that 0 (the button knot) and 1 (the top knot) are multiple knots. In fact, if the curve is of degree $p$, the multiplicity of 0 and 1 should be $p+1$.

If you choose **With Uniform Knots**, the curve will not be tangent to the control polygon and every knot is a simple knot. This is shown in the above middle figure.

Finally, if you choose **With Closed Knots**, the curve becomes a closed loop and some knots are shown with a different color. Knots marked with blue triangles are normal knots, while those non-blue triangles marked special knots that cannot be manipulated.

You can change the curve to another form at any time. Just select **Curve**, followed by **Show Curve Segment**, followed by the desired form.
Tracing the Curve - de Casteljau's and de Boor's Algorithms

Tracing a curve is a simple task. Dragging the small triangle (i.e., the \(u\)-indicator) in the right-hand side of the vertical slider moves its corresponding point on the curve. This point is referred to as the *tracing point*.

De Casteljau's Algorithm

One can activate de Casteljau's algorithm for tracing a curve. De Casteljau's algorithm is an elegant algorithm for computing the corresponding point on a Bézier curve of a given \(u\). As shown in the following figure, activating button de Casteljau's Alg. in the Display & Tracing Options Window displays all intermediate computation steps of the de Casteljau net:

![Image of de Casteljau's algorithm](image)

The above only displays the de Casteljau net graphically. This system can also show the detailed computations step-by-step. To see this, select Window followed by Triangular Computing Scheme. This will bring up the Triangular Computing Scheme Window. This window has a button STEP for displaying the computation of de Casteljau's algorithm step-by-step. Clicking on STEP displays the left-most column which contains all given control points as show below:
Clicking on **STEP** again displays the second column, each element of which is computed as the combination of the north-west and south-west control points on the first column. The south-east pointing and north-east pointing arrows have values $1-u$ and $u$, respectively. Note that the points on each column correspond to a polyline of the de Casteljau net. Since we have two columns, we have two polylines of the de Casteljau net.

Clicking on **STEP** twice produces the following result. As you can see, we have four columns and hence four polylines of the de Casteljau net.
Clicking on **STEP** twice again will produce a single point, which is the point on the curve corresponding to the given $u$.

**Button** **RESET** resets the computation so that we can restart the computation.

The vertical and horizontal sliders in the **Triangular Computing Scheme Window** are used for scrolling the triangular-shaped computation vertically and horizontally, respectively.

**De Boor's Algorithm**

**De Boor's algorithm** is an extension of de Casteljau's algorithm. De Boor's algorithm can be used on all four types of curves and when it is applied to a Bézier curve, it reduces to de Casteljau's algorithm. Note that in the computation of de Boor's algorithm, unlike de Casteljau's, not all control points are involved. As a matter of fact, if the degree of a B-spline (or NURBS) curve is $p$, there are no more than $p+1$ control points involved. The following shows a B-spline curve of degree 5 and its convex hull and de Boor net at $u=0.14$. Since its degree is 5, only six control points are involved. Since $u=0.14$, the involved control points are from 0 to 5.

Another major difference between de Casteljau's and de Boor's algorithm is that the weights for computing intermediate control points of the de Boor net are not $1-u$ and $u$. In the case of de Boor's algorithm, these weights must be computed separately as they are different in each stage. As a result, the new points do not subdivide the line segments in a ratio of $u/(1-u)$.
Selecting Window followed by Triangular Computing Scheme brings up the Triangular Computing Scheme Window. This window works for both de Casteljau's and de Boor's algorithms. However, when the curve is a B-spline or NURBS, only those involving control points will be shown. Clicking on STEP will display the first column which is actually part of the control polygon.

Clicking on STEP once computes the second column:
Clicking on **STEP** two more times finishes columns 3 and 4.

Clicking on **STEP** twice again reduces the number of points to one and this is the point corresponding to \( u = 0.14 \) on the B-spline curve.
On-the-Fly Computation

One of the many nice features the curve system provides to you is that you can drag the $u$-indicator to trace the curve and at the same time see the change of convex hull, de Casteljau or de Boor net, and the triangular computation scheme for obtaining the corresponding point on the curve. For example, if the $u$-indicator moves to $u = 0.60$, you will see a continuous update of the tracing point, convex hull, de Casteljau or de Boor net, and the triangular computing scheme. The following figures show the situation at $u=0.6$. 

There are three ways of modifying the shape of a curve, namely modifying control points, modifying knots, and modifying the weights of control points. The first works for all four types of curves, the second can be applied to B-spline and NURBS curves, and the third can be used with rational Bézier and NURBS curves. This page concentrates on the first one: modifying the positions of control points.

To modify control points, we can move the existing control points or add new or delete existing ones.

**Moving Control Points**

Moving control points can be used with all four type of curves. There are two ways to move control points in the $x$- and $y$-directions. To move control points in space, please refer to Moving into Space for the details.

To move a control point, select Edit followed by Move. This puts the system in the move mode and the cursor becomes a north-west pointing arrow. Then, you can click and drag a control point to move it to another place. Note that when you click and drag a control point, it becomes selected and is shown in a different color. Once you release the mouse button, the new coordinates and weight of that control points will be shown in the X, Y, Z and W buttons.

The following figure shows a B-spline curve of degree 4. We first enter the move mode by selecting Edit followed by Move. Then, click on the control point with ID 10. Its coordinates and weight will be shown. In this case, as shown in the following figure, the coordinates of this control point is (44.0, 85.0, 0.0) with weight 1. Note that since this is a B-spline curve, the weight of any control point must be 1. Note also that since this is a curve on the $xy$-plane, the $z$-coordinate has value zero.

If control point 10 is dragged to a new position as shown in the figure below, its new coordinate values will be updated in the X, Y, Z and W buttons. The new coordinate becomes (67.5,-5.5,0) with weight 1. It is important to notice that when a control point of a B-spline or a NURBS curve moves, only portion of the curve will be affected. This can be seen by comparing the figures above and below. This is the well-known property known as the local modification property and is one of the many advantages of using B-spline and NURBS curves. Modifying the position of a single control point of a Bézier or a rational Bézier curve will change the shape of the curve globally.
After a few experiments, you will find out that moving a control point will pull the curve in the direction of movement. Note also that moving a control point does not change the specification of the curve (i.e., the number of control points, the number of knots and the degree of the curve). Only the shape of the curve changes.

Deleting a Control Point

Since deleting and inserting control points change the original specification of a curve, use it with care.

The above is a B-spline curve of degree 4 used earlier in the discussion of moving control points. The point marked with a square is control point 10 and the point marked with an ellipse is the tracing point on the curve corresponding to $u = 0.55$.

To delete a control point, select **Edit**, followed by **Delete**. The cursor changes to a skull, indicating that the system is in the **delete** mode. Move this delete (or skull) cursor on top of control point 10 and click. Then, the curve, its control polygon and convex hull all disappear. The control point you just clicked on also disappear and all remaining control points are re-numbered. All control points before the deleted one keep the original control point IDs, and all IDs of the remaining control points will be
decreased by one. Moreover, the control point with the new ID 10 becomes selected. This is shown below.

Since deleting a control point has ruined the original curve specification, this system considers you are creating a new one. What you need to do is to display the new curve. Select Curve, followed by Show Curve Segment, followed by With Uniform Knots, With Clamped Knots or With Closed Knots. The new curve will be displayed. The following figure shows the new curve with clamped knots.

Please note that the convex hull for $u=0.55$ changes. But, the curve itself does not change very much. One can easily notice that the right bulge becomes smaller and the left one of the curve has no change at all. This is due to the local modification property of B-spline and NURBS curve.

Deleting control points from a Bézier or rational Bézier curve is simpler. One can just delete a control point and due to the impact of deleting a control point on Bézier and rational Bézier curves being global, the system will immediately display the new curve and adjust its degree. The right figure below is the result of deleting control point 10 from the left one.
Inserting New Control Points

Since inserting control points, like deleting control points, will also change the original specification of a curve, use it with care.

Since control points are ordered, one must specify where the new control points should be placed. This system provides two options Insert After and Insert Before. The following figure shows a B-spline curve of 6 and two more control points will be inserted after control point 3.

To insert new control points after a control point, one need to do two steps: (1) indicating the control point after which new control points will be inserted (i.e., in this case, one should indicate control point 3), and (2) entering the insert mode.

To carry out the first step, if the system is already in move mode, one can click on a control point (i.e., in this case, clicking on control point 3). If the system is not in the move mode, one can either enter the move mode or select Edit followed by Select to enter the select mode. Then, click on the control point you want to select. The difference between the move mode and the select mode is that the former selects a control point and allows you to move it, while the latter only allows you to select a control point.

After a control point is selected, one can select Edit followed by Insert After for inserting new control points after the select.
one, or **Insert Before** for inserting new control points before the select one. Let us try **insert after** first.

Select **Edit** followed by **Insert After**. The cursor changes to a cross, meaning that the system is in the **insert** mode and the menu button displays **Ins Aft**. Then, each click on the drawing canvas adds a new control points after the selected one. If the selected one is control point \( k \), the new control points will be \( k+1 \), \( k+2 \) and so on. However, since inserting new control point changes the curve specification, the curve will disappear as we saw in the case of deleting control points. The following shows the result of adding two new control points after control point 3:

To get the curve back, we need to select **Curve**, followed by **Show Curve Segment**, followed by **With Uniform Knots**, **With Clamped Knots** or **With Closed Knots**. The new curve will be displayed. The following figure shows the new curve with clamped knots.

Based on this idea, the **create** mode is actually equivalent to inserting new control points **after** an imaginary control point, say control point -1.

The process for inserting new control points **before** a select one is similar. If the selected control point is \( k \), after inserting a new control point **before** control point \( k \), the new control point becomes \( k \) and the original \( k \) and all control points after \( k \) will be
increased by one.

Note that *insert after* and *insert before* could have a dramatically different effect even though new control points are inserted at the same locations. The following left figure shows inserting two new control points *before* control point 3. The positions of the new control points are near to those two we used in *insert after*. After insertion, the new control points become 3 and 4 and the original control point 3 changes to control point 5. Thus, the curve will have a twist as shown in the right figure.
Modifying Weights

There are three ways of modifying the shape of a curve, namely modifying control points, modifying knots, and modifying the weights of control points. The first works for all four types of curves, the second can be applied to B-spline and NURBS curves, and the third can be used with rational Bézier and NURBS curves. This page concentrates on the third one: modifying the weights of control points.

Since only rational Bézier and NURBS curves allow control points to have weights, modifying weights can only be used with these two types of curves. To modify the weight of a control point, this control point must be selected first. If the system is in the move mode, clicking on a control point selects it. Otherwise, one can enter the move (resp., select) mode by selecting Edit followed by Move (resp., Select). Then, clicking on a control point selects it.

After selecting a control point, its color changes and its coordinates and weight will be displayed in the four buttons near the bottom of the drawing canvas. Clicking on $W$ selects changing weight. The button color will change to blue meaning that $W$ is selected for further editing. Finally, use the Translate slider below the $W$ button to adjust the value of the selected weight. As the weight changes, the shape of the curve changes on-the-fly.

Let us take a look at an example. The following shows a NURBS curve of degree 6 with control point 8 selected.

Now, click on $W$ to select weight. Then, increase its value to 3 with the Translate slider. The curve will be pulled toward the selected control point. This is shown in the left figure below. If the weight is further increased to 10, the curve will be pulled further toward the selected control point. In fact, increasing the weight of a control point pulls the curve toward that control point. Moreover, if the weight is increased to infinity, the curve passed through the selected control point.
Let us decrease the weight to 0.5. The curve is pushed away from the selected control point as shown in the left figure below. If the weight is decreased further, the curve is pushed further away. The right figure below has the weight decreased to 0.1.

Decreasing the weight of control point pushes the curve away from that control point.

What if the weight is changed to zero? This removes the impact of the selected control point. The left figure below has weight zero for control point 8. The affected part of the curve becomes flat meaning that the influence of control point 8 disappears. If the weight decreases further becoming a negative value, the curve will move further away from the selected control point as shown in the right figure below. In this case, the curve segment moves outside of its convex hull, violating the convex hull property.
The convex hull property helps us determine the bound of a curve segment. This can be achieved only if the weights of control points are all non-negative. As a result, even though this system supports negative weights (the right figure above), it is usually unwise to use them.
There are three ways of modifying the shape of a curve, namely modifying control points, modifying knots, and modifying the weights of control points. The first works for all four types of curves, the second can be applied to B-spline and NURBS curves, and the third can be used with rational Bézier and NURBS curves. This page concentrates on the second one: modifying the knots of a B-spline or a NURBS curve.

Before discussing knot modification, we need to review the drawing canvas. The vertical slider is for curve tracing and displaying knots and their multiplicities. The small triangle in the right-hand side is referred to as the $u$-indicator. Dragging the $u$-indicator up and down moves the corresponding point, the tracing point, on the curve. The figure below shows the $u$-indicator at $u=0.2$ and its corresponding point. Both are marked with black rectangles.

The left-hand side of the slider has the knots and their multiplicities. Each knot is displayed in a form of $m \ (d.\ddd)$, where $d.\ddd$ is the value of a knot and $m$ is its multiplicity. The positions of knots are marked with little triangles in the left side of the vertical slider. These triangles are called the knot indicators. The figure above has 16 knots. They are 0 (multiplicity 7), 1/3 (simple knot), 2/3 (simple knot), and 1 (multiplicity 7). Knots that are not 0 and 1 are internal knots. Each knot has its corresponding point on the curve, shown as smaller spheres. These are called knot points. In the figure above, knot 1/3 and its corresponding point on the curve are marked with blue rectangles, and knot 2/3 and its corresponding point on the curve are marked with blue ellipses.

Knot points subdivide the curve into curve segments, each of which is a curve of degree the same as the given one. In the figure above, there are three curve segments obtained from the two knot points at both ends (i.e., knot points corresponding to 0 and 1) and two knot points corresponding to the internal knots. Since a knot may be a multiple knot, the number of knot points is always less than or equal to the number of knots.

Moving Knots

Changing the value of a knot will change the shape of the curve. But, it is not clear and perhaps even unpredictable about the relationship between the change of the shape and the change of a knot. As a result, changing the value of a knot may not provide a satisfactory way of shape modification.

To change the value of a knot that is not 0 and 1, one can drag the corresponding knot indicator shown in the left-hand side of the vertical slider. As the knot moves, the shape of the curve changes on-the-fly. In the following figure, the value of $u$ is 0.2. If the knot 2/3 is moved to 0.8, the corresponding knot point moves closer to control point 6, and the curve segment between this knot point and the end point becomes shorter (see the left figure below). At the same time, the other knot point moves as well. It moves toward control point 1, making the curve segment between these two knot points longer than the original (see the figure above). The curve segment between the knot point corresponding to 2/3 and the end point has a sharper turn.
Moving knot 1/3 to 0.1 yields the same effect as described earlier (see the right figure above).

Creating Multiple Knots

A knot can be moved on top of another, creating a multiple knot. In this case, the two corresponding knot points coincide. The following figure shows a B-spline curve of degree 3 with multiple knot 0 (multiplicity 4), simple knots 0.25, 0.5 and 0.75, and multiple knot 1 (multiplicity 4). We intend to move knot 0.25 on top of knot 0.5, making 0.5 a multiple knot of multiplicity 2.

As knot 0.25 moves, you will see the knot points corresponding to 0.25 and 0.5 move toward each other and eventually become the same knot point when knot 0.25 is moved on top of knot 0.5. Moreover, the curve is tangent to the line segment of control point 2 and control point 3 at the knot point corresponding to 0.5. This is shown in the left figure below.
Now, let us move the other knot 0.75 to the new double knot 0.5, making it a triple one. As knot 0.75 moves, the two knot points move toward each other and at the same time move toward control point 3. Once knot 0.75 moves on top of knot 0.5, the two knot points and control point 3 become identical. Moreover, the curve at control point 3, which now corresponds to knot 0.5, is no more smooth (i.e., the tangent vectors at control point 3 of each segment are different. This is shown in the right figure above.

In theory, we know that a B-spline or a NURBS curve at a knot of multiplicity \( k \) is of \( C^{p-k} \) continuity, where \( p \) is the degree of the curve. Therefore, in the right figure above, the knot point corresponding to triple knot 0.5 is \( C^0 \). That is, the tangent vectors of the two sides at that knot point are different.

The curve is tangent to a line segment at a knot point, if the multiplicity of the corresponding knot is \( p-1 \), where \( p \) is the degree of the given B-spline or NURBS curve. Note that the curve is of \( C^1 \) continuity at that knot point (i.e., the tangent vectors at that point of the curve segments are equal). The line segment to which the curve is tangent depends on the location of the knot and is determined using the convex hull property. As a result, in the left figure above, since 0.5 is a double knot and the curve is of degree 3, the curve is tangent to a line segment of the control polygon.
Working with drawing canvas only allows you to create curves in the \(xy\)-plane. To design space curves, we need to work with the \(z\)-axis and use transformations that can be applied to all three coordinate axes.

All previous discussions only show the curve on the \(xy\)-plane. The \(z\)-axis is the axis that is perpendicular to the \(xy\)-plane (i.e., the screen). To see the \(z\)-axis, space transformations are required. This system supports transformations (i.e., translations and rotations) applied to the whole scene or to a curve. Transformations applied to the scene will transform the coordinate axes as well as all curves. One might consider scene transformations equivalent to moving one’s camera around. Transformations applied to a curve only translate or rotate the current curve, the curve selected for further editing. The coordinate axes remain unchanged.

Space transformations are performed using buttons and sliders in the bottom part of the drawing canvas as shown below:

Clicking on Curve and Scene selects the current curve and the scene to be translated or rotated, respectively. Clicking on Point only allows the selected control point to be translated. The ID of the selected control point is shown in the Point button and the coordinate values and weight are shown in buttons \(X\), \(Y\), \(Z\) and \(W\). The top slider is for translation and the bottom slider is for rotation.

### Transforming the Scene

To move a planar curve into space, the first step is perhaps rotating the scene followed by translating some selected control points in the \(z\) direction. Note that the \(x\)-axis and \(y\)-axis point to the left and top of the screen, respectively, while the \(z\)-axis points to the viewer (i.e., right-handed system).

This system has a virtual and invisible world coordinate system which is identical to the one shown on the drawing canvas when the scene is not translated and rotated. If you select Techniques followed by Show Coordinate Axes, you will see the coordinate axes. The horizontal one is the \(x\)-axis, the vertical one is the \(y\)-axis, and the one pointing at you is the \(z\)-axis.

To translate or rotate the scene, we should first select what should be transformed (i.e., scene or curve), followed by a reference axis (i.e., \(x\)-, \(y\)- or \(z\)-axis), followed by moving the little triangles of the two sliders to generate the desired effects. The following is a B-spline curve of degree 5 defined by 12 control points:
Now we want to rotate the scene about the y-axis. Clicking on the **Scene** button. It will be shown in blue color indicating that all subsequent transformations will be applied to the whole scene. To carry out rotation about the y-axis, click on button **Y**. Then, it is also shown in blue, indicating that all subsequent transformations will be performed with respect to the y-axis (i.e., translating in the y-axis direction and rotating about the y-axis). Finally, sliding the little triangle of the **Rotate** slider will rotate the scene about the y-axis, and sliding the little triangle of the **Translate** slider will translate the scene parallel to the y-axis. The following figure shows the result of rotating the scene about the y-axis:

![Figure showing rotation about the y-axis]

Next, let us rotate the scene about the x-axis. **Please note that a scene rotation is always about one of the three virtual and invisible coordinate axes.** The x-axis is the one parallel to your screen's horizontal edge, the y-axis is the one parallel to your screen's vertical edge, and the z-axis is perpendicular to your screen. These three *virtual* axes never change. In this way, you will have a world coordinate as an absolute reference. To rotate the scene about the x-axis, click on **X** would be sufficient because the button **Scene** is still active. The following shows the result of rotating the scene about the (virtual) x-axis. Now you can see all three coordinate axes and the position of the curve (in the xy-plane):

![Figure showing rotation about the x-axis]
Moving Control Points

As you can see from the above figure, the curve is planar in the \( xy \)-plane. Since the control points are in space, we can move them out of the \( xy \)-plane. To do so, you need to select the control point by clicking on it. Then, the coordinates and weight of the selected control point are shown in the buttons \( X, Y, Z \) and \( W \). Also, the ID of the selected control point will be shown in the button \textbf{Point} in blue color, indicating that all subsequent translations will be applied to this point. Note that control points cannot be rotated. To translate control points in space, one can only use the \textbf{Translate} slider. Click on \( X, Y \) or \( Z \) to choose a translate direction, followed by using the \textbf{Translate} slider to position the selected control point. To translate another control point, select it and repeat this procedure. The following figure shows the result of translating some control points in the \( z \)-direction. The result becomes a space curve rather than a planar one. Click here to download the file (\texttt{space-curve-1.dat}) that define the above space curve.

The following figure shows the \( xy \)-plane view of this curve, which looks exactly the same as the original planar curve since we did not move any control points in the \( xy \)-direction.

If we rotate the scene so that the \( x \)-axis is perpendicular to the screen, the result is the left figure below. It is obvious that now all control points have \( z \)-coordinate different from 0. The right figure below is the result of rotating the scene so that the \( y \)-axis is perpendicular to the screen.
Transforming the Current Curve

If button **Curve** is selected, then all transformations are applied to the current curve and the scene will not change. However, the reference coordinate axis for translation and rotation are the coordinate axes shown on the drawing canvas rather than the virtual world coordinate axes as used for transforming the scene. The following figure shows the result of rotating the above curve about the z-axis.

Note that you can intermix scene and curve transformations to complete your design. However, at any time, a transformation can either be applied to the selected point, the current curve or the scene.

**Zooming**

Zooming is equivalent to moving a camera toward (*i.e.*, zooming in) or away from (*i.e.*, zooming out) the object. Thus, **zooming can only be applied to the scene**. Clicking button **In** (*resp.*, **Out**) moves the camera toward (*resp.*, away from) the scene. Sometimes, after a couple of transformations, part of the scene may be moved outside of the drawing canvas. In this case, zooming in/out would be very useful.
The derivatives of Bézier and B-spline curves are still Bézier and B-spline curves, respectively. In fact, the derivatives of a Bézier (resp., B-spline) curve is a curve defined by a new set of control points that are constructed from the original set of control points. The degree of the derivative curve is, of course, one less than the original. The following figure shows a B-spline curve of degree 5 with tracing point corresponding to $u = 0.6$.

To display the first derivative curve or hodograph, select Window, followed by Hodograph (First Derivative). To display the second derivative curve, select Second Derivative. Then, a new window appears showing the first or second derivative curve in the same type of the current curve on the drawing canvas. The left (resp., right) figure below shows the hodograph (resp., second derivative) curve of the above B-spline curve. As a result, both derivative curves are B-spline curves.

Each of these derivative windows contains three buttons:

1. **Magnify** zooms in and makes the derivative curve looks larger.
2. **Shrink** zooms out and makes the derivative curve looks smaller.
3. **Dismiss** closes the window.
Since the size or range of the curve is usually unpredictable before computation starts, buttons **Magnify** and **Shrink** help make the derivative curves fit into the window boundary. Once these derivative windows are shown, moving the $u$-indicator will not only move the tracing point on the curve, but also the tracing points on each of the two derivative curves.

Unfortunately, the derivative curves of a NURBS (and hence a rational Bézier) curve is in general not a curve in one of the four types. As a result, no control points will be shown in the derivative windows. These derivative curves are approximated with line segments. The following figure shows a NURBS curve of degree 5.

The left (resp., right) figure below shows the hodograph (resp., second derivative) curve. You can see clearly that the curves are made from line segments and no control points are shown.
The Moving Triad

The moving triad is a local coordinate system with origin at a point on the curve. The three "coordinate axes" are the tangent vector, the binormal vector and the normal vector. The tangent vector gives the direction of movement of the tracing point, the normal vector points the direction the curve is turning, and the binormal vector points to the "sky" and is defined to be the vector perpendicular to the tangent and normal with the right-hand system.

This system is capable of displaying the moving triad and as the tracing point moves on the curve the moving triad moves as well. Note that the moving triad should be viewed in space; otherwise, you would only see the tangent and normal vectors because the binormal vector is perpendicular to the screen. Therefore, scene rotation may be required.

To display the moving triad, bring up the Display & Tracing Options Window and click on button **Tangent, Binormal & Normal**. The three vectors will be shown in different colors. In the following figure, the tangent vector is shown in red, the binormal in green and the normal in blue. Note that you can always customize the colors, and as a result, the colors on your system may be different from what are being used on this page. Click [here](#) to download a copy of this file `space-curve-1.dat` for your practice.

The following is another triad at a different location. In many cases, it is not entirely clear where these vectors are pointing. Thus,
scene rotation may be required to clearly see the directions of these three vectors.
The Curvature Sphere

The curvature sphere provides information of the turning rate at a point on the curve. The center of the curvature sphere is always on the positive direction of the normal vector. The curvature at a point on the curve is the reciprocal of the radius of the curvature sphere. As a result, the larger the curvature sphere, the smaller the curvature and the flatter the curve. If a curve has a very sharp turn at a point, the curvature at that point must be very large and the curvature sphere at that point will be very small. On the other hand, if a curve is flat at a point, the curvature is very small and the curvature sphere will be very large. Consequently, a straight line has curvature zero because there is no turn on a line, and the curvature sphere will have radius an infinite value at any point on a line. Based on this discussion, it is clear that examining curvature spheres will help you understand if a curve has sharp turns.

To display curvature sphere, you must first activate **Tangent, Binormal & Normal**. Once the triad is displayed, click on **Curvature Sphere** will display the curvature sphere.

The curvature sphere moves according to the movement of the \( u \)-indicator. As mentioned earlier, the center of the curvature sphere is on the positive direction of the normal vector, and is displayed as a square. The left figure below shows a curvature sphere. But, when the tracing point moves to the top of the curve which is flat, the curvature sphere becomes bigger indicating the curve is flat in the vicinity of the tracing point.

If the tracing point moves further to an inflection point (i.e., a point whose second derivative is zero), the curvature sphere becomes extremely large. Moving pass this inflection point and entering a sharp turn, the curvature sphere is small, meaning that the curvature is large and the curve has a sharp turn there.
Imagine that you are sitting on a roller coaster box following the track. The track can be considered as a curve. Thus, the box moves in the direction of the tangent vector at the current position, the binormal vector points where your head points, and the normal vector points to the turning direction. More precisely, if you have a left (resp., right) turn, the normal vector points to the left (resp., right). As your box follows the track, you are sitting at the tracing point of a curve and have a "local" view of the world. This system can provide you with such vivid effect.

Before activate this effect, make sure that **Tangent, Binormal & Normal** has been activated. If you want to see the curvature sphere, you can also activate **Curvature Sphere**. Then, click on **Roller Coaster** to activate the roller coaster effect.

In the following, we still use file **space-curve-1.dat** as our working example. Click **here** to download a copy. After clicking on **Roller Coaster**, the **Roller Coaster Window** appears. It shows the curve, the tracing point, and the moving triad. The eye position can be modified to obtain a better view. We shall return to this later on. The left figure below shows the tracing point at the beginning of the curve. The roller coaster window also depicts the same. We can actually see control point 0 behind the tracing point.

Let us move the tracing point further entering a sharp turn. Now we can see control point 3 in front of the tracing point; but our
position is somewhat up-side-down because the binormal vector (the green one) points downward. However, since the binormal vector points to where our head is pointing, even though the curve goes up, the roller coaster window shows a down turn. This matches our roller coaster experience.

Let us move forward to a place just before entering an inflection point as mentioned in the Curvature Sphere page. From the roller coaster window, it is clear that the curve is about to change its turning direction.

The following figures show the result after passing the inflection point. Please notice that the curvature sphere changes sides as well.
It may be difficult for you to extrapolate your roller coaster experience on a screen. But, going around the curve a few more times and, if it is necessary, do some scene rotations, you will be able to see and learn more about the effect of moving triad and curvature, and gain some visual experience of how curve turns!

### Adjusting Eye Position

Clicking on the **Eye Control** button of the *Roller Coaster Window* brings up the *Roller Coaster Eye Position* window (see below). This window contains buttons for changing the viewing position (*i.e.*, your seat) in the roller coaster box. One can move near or farther from the center (*i.e.*, the tracing point), left or right, and up or down.

There are two buttons for choosing the up direction. It is either based on the binormal vector or the negative normal vector. Normally, since the binormal vector points to the head direction, the up direction, by default, is the binormal direction. For interesting effect, one can choose the negative direction so that you would get an interesting viewing effect. Click on button **Binormal** (*resp.*, **Negative Normal**) to use the binormal (*resp.*, the negative normal) direction as the up direction. The left figure below is the normal case, the up direction being the binormal direction, while the right one uses the negative normal direction as the up direction.
There are three sliders for modifying the eye position. Please note that wherever the position, the passenger always look at the tracing point. In the following, we use the binormal vector as the up direction. We can move closer to or farther away from the tracing point with the Near/Far slider. The default value is 0.2. The left figure below shows moving closer to the tracing point (0.25), while the right one shows moving away from the tracing point (1.0).

We can also use the Left/Right slider to move to the left or right. The default value is that the view position is slightly shift to the right side at a value of 0.1. If we move to the left to -0.1, we are in the "left" side of the curve. This is shown in the left figure below. The right one shows the effect of moving to the right side of the curve at 0.25.

Use the Low/High slider to move up and down. The default value is 0.2, meaning that the viewing position is slightly above the floor of the roller coaster box. The left figure below shows a value of 0.05, which is near to the floor of the roller coaster box and as a result the tangent vector can barely be seen. The right figure shows the effect of moving up to 0.5, which is high above the floor.
The Roller Coaster Effect

Changing the Color Scheme

As you perhaps have noticed that colors used on these pages are different from those on your system. This is because you can change the colors of many displayable items.

To change colors, select Window, followed by Change Color Scheme. This will bring up the Color Scheme Window as shown below.

In this window, you will see four sliders: RED, GREEN, BLUE and ALPHA. The first three sliders are for changing the intensity level of the three color components (i.e., read, green and blue). The fourth slider controls the level of transparency of a color (i.e., the alpha channel). More precisely, alpha channel value 0 (resp., 1) produces a completely transparent (resp., non-transparent) color. An alpha channel value closer to 0 produces a color more transparent than an alpha value closer to 1.

The button below Choose a color object will cause a pull-down menu to appear, which contains all objects whose colors can be altered. These objects include the drawing canvas, the current curve, control point, control point index, the selected control point, the tracing point, knot points and so on. You can also change the color used for displaying the de Casteljau or the de Boor net. There are only ten levels of colors, named as De Casteljau 1 to De Casteljau 10.

To change the color of any one of these objects, select that object from the pull-down menu, followed by changing each of the color components and their alpha channel value using the four sliders. The color corresponding to the current setting will be shown in the small square. If you are satisfied with a particular color, clicking on APPLY will apply the selected color to that object on the drawing canvas. Please note that the color you see on the drawing canvas may be slightly different from the color shown in the color square since objects on the canvas are illuminated with light sources.

This process can be repeated as many times as you want until you are satisfied. Note that the convex hull should have a transparent color so that the control polygon, control points and other objects can be seen through the convex hull. Otherwise, a non-transparent convex hull will block almost all the details of a curve.

Saving Your Color Changes

The color changes made with the Color Scheme Window can be saved. To save the color changes, select File followed by Save Color Scheme. This will save the new color scheme to a file curve.rc in the current directory. Next time when you run the curve system, it will search your current directory for this file and initialize itself with the color scheme saved there. If this file is not found, a default color scheme will be used. So, if you are not satisfied with the saved color scheme, you can remove file curve.rc or use another directory as the current directory to run this program.

There is a second way to keep track the color scheme being used. Start the curve system and save the color scheme before making any color changes. This will create a file curve.rc. Then, you can rename this file as curve.rc.default. When it is necessary to go back to default colors, just rename it to curve.rc.

Although you can change the color of the current curve, since you can make any curve the current one, the color for the current curve is not well-defined. As a result, saving the color scheme does not guarantee the color for the curve will change next time when the system runs. More precisely, if you set the current curve to use color red and save the color scheme, then when you run the curve system again the curve may not be shown in red.

An improvement will be made in a future version.

Knot Insertion

Knot insertion is a fundamental algorithm in curve and surface design. Its goal is to add one or more knots to the knot vector without changing the shape of the curve. Due to the fundamental identity $m = n + p + 1$, where $m+1$, $n+1$ and $p$ are the number of knots, the number of control points, and the degree of a B-spline/NURBS curve, increasing the number of knots causes the number of control points to be increased the same amount. Thus, the existing set of control points must be modified to accommodate the increase of knots.

Although one can insert several knots at the same time, this system chooses to do it one-by-one for you to see and understand the effect of knot insertion. Follow the steps below to insert a knot:

1. Move the $u$-indicator to a place where you want to insert a new knot. This new knot can be identical to an existing knot, and in this case increase the multiplicity of the existing knot by one. If the new knot is not identical to an existing knot, after insertion, this becomes a simple knot.

2. Select Techniques followed by Knot Insertion. Then, a new knot indicator at the place of the $u$-indicator appears and at the same time the existing set of control points will be modified.

The following is a NURBS curve of degree 4 defined by 10 control points. We want to insert a new knot at $u = 0.85$. Since 0.85 is not an existing knot, the tracing point is not on any knot point.

Selecting Techniques followed by Knot Insertion yields the left figure below. As you can see, we have one more knot at $u = 0.85$ and the number of control points is increased by one. If we put the new and the original control polygons together, as shown in the right figure below where the new control points are marked with blue rectangles, you will see that the polyline defined by control points 5, 6, 7, 8 and 9 is modified by cutting the corners at 6, 7 and 8. In this way, one new control point is created.
The following is a NURBS curve of degree 5 defined by 12 control points. This NURBS curve has knots 0 (multiplicity 6), 0.4 (multiplicity 3), 0.55 (simple), 0.7 (multiplicity 2) and 1 (multiplicity 6). We want to insert a new knot at 0.7, making knot 0.7 a triple knot after the insertion.

Move the \( u \)-indicator to 0.7 and select **Techniques** followed by **Knot Insertion**. This gives the left figure below. Now the curve is defined by 13 control points and those control points near the tracing point at \( u = 0.7 \) are modified. To see the modification, let us put the new and the original control polygons together as shown in the right figure below. You should see that the affected polyline is defined by the original control points 6, 7, 8 and 9, and that the corners at 7 and 8 are cut to create the three new control points marked by blue rectangles.
In general, if $p$ is the degree of a B-spline/NURBS curve, $s$ is the multiplicity of the new knot before insertion, and $c$ is the number of corners to be cut, then we always have $p = s + d$.

**How to overlay two curves and their control polygons**

You can save the first curve and then work on the second. Finally, import the first curve back on the canvas.

**Further Effects of Knot Insertion**

Knot insertion is a very useful algorithm that can be used in many applications, including an implementation of de Boor's algorithm. Here, we shall show you some important facts.

The left figure below is a NURBS curve of degree 4 defined by 8 control points. The figure also shows the tracing point corresponding to $u = 0.9$, which is not a knot. Click here to download a copy of this scene to file knot-insert.dat. If $u = 0.9$ is inserted as a new knot, three corners are cut and the result is shown in the right figure below.

If $u = 0.9$ is inserted again, it is inserted at an existing knot, making 0.9 a double knot. The result is shown in the left figure below. If 0.9 is inserted the third time, this will make 0.9 a triple knot and the result in the right figure below suggests that line segment of control points 6 and 7 is tangent to the curve at the tracing point. In fact, if $p$ is the degree and $u$ is a knot of multiplicity $p - 1$, then the curve is tangent to a line segment of the control polygon at the point on curve corresponding to $u$.
If \( u = 0.9 \) is inserted the fourth time, the point on the curve corresponding to the knot becomes a control point (i.e., it has a control point index)! More precisely, if \( u \) is inserted so that the multiplicity after insertion is equal to the degree of the curve, then the point on the curve corresponding to \( u \) is a control point. That is, the curve passes through one of its control points. In fact, de Boor's algorithm is implemented this way. Given a \( u \), to find its corresponding point on the curve, we can repeatedly insert \( u \) until its multiplicity is equal to the degree of the curve. Then, the last control point is the desired point that corresponds to this \( u \). This is shown in the figure below.

**Knot Splitting**

It is impossible to remove a knot without changing the shape of the curve. Therefore, this system implements "knot splitting" for splitting a multiple knot. However, the shape of the curve will change slightly.

You can move the \( u \)-indicator to a multiple knot. Then, select **Techniques** followed by **Knot Splitting**. The multiple knot will be split into two. If the multiple knot is of multiplicity \( k \), after splitting the multiple knot will have multiplicity \( k - 1 \), and a new simple knot will be created nearby. Of course, the shape of the curve will change; however, if the newly created simple knot is moved to its "parent" multiple knot, the curve will return to its original shape.

The left figure below is a NURBS curves of degree 5 with knots 0 (multiplicity 6), 0.4 (multiplicity 3), 0.55 (simple knot), 0.7 (multiplicity 2) and 1 (multiplicity 6). This curve was used earlier on this page. Click here to download a copy of this file knot-split.dat. The \( u \)-indicator is at \( u = 0.4 \), the first internal knot. Its tracing point is also shown. Selecting **Techniques** followed by **Knot Splitting** yields the right figure below. The triple knot 0.4 is split into two, one double knot at 0.4 (the original place) and one simple knot 0.38. It is hard to see the difference between the following two curves. However, if you import the original file into the scene after the triple knot is split and use different colors for the curve segments, you should be able to see
some very subtle difference between the two curves. Thus, unlike knot insertion, knot splitting does not preserve the shape of the curve.
**Degree Elevation**

*Degree elevation* increases the degree of a curve *without* changing the shape of the curve. Together with *knot insertion*, they provide two frequently used techniques to make two curves *compatible*. Two curves are compatible if they are of the same type and have the same number of control points, the same number of knots and the same degree. Due to the fundamental identity, if the degree and the number of knots are known, the number of control points are determined. As a result, if two curves of the same type have the same number of knots and the same degree, they much have the same number of control points. Therefore, knot insertion is used to make the knot vectors identical and degree elevation makes the curves of the same degree.

Select **Techniques** followed by **Degree Elevation**. The degree of the current curve will be increased by one. To increase the degree by more than one, repeat this process.

The left figure below is a Bézier curve of degree 14. After performing degree elevation increasing its degree to 14, we have the right figure below. It now has 15 control points. Note that the shape of the Bézier curve does not change.

Degree elevation for B-spline and NURBS curves is performed by subdividing the curve into Bézier curve segments, increasing the degree of each Bézier segment, and combining them together back to a single B-spline and NURBS curve. So, increasing the degree of a Bézier curve is an important procedure.

The following left figure is a NURBS curve of degree 4 defined by 8 control points. After degree elevation, we have the right figure. It is of degree 5 defined by 12 control points.

If you are careful, you may have already found out that degree elevation will cause the set of control points modified "globally".
More precisely, unlike knot insertion, all control points, except for the two endpoints, are replaced with new ones.
**Curve Subdivision**

*Curve subdivision* subdivides a curve at a point so that each of the two components possesses the same type. Follow the procedure below to perform curve subdivision.

First, move the \( u \)-indicator to a place where you want to cut the curve. Select **Techniques** followed by **Curve Subdivision**. The original curve will be cut into two at the specified point.

Under this system, if a curve is subdivided at a particular \( u \), the first curve segment has domain \([0,u]\) and the second curve segment has domain \([u,1]\). After subdivision, the current curve is always the second one. The curve and its domain have the same color. You can always change the colors. See **Changing the Color Scheme** for the details.

In the following figure, \( u \) is placed at 0.8. We intend to cut the curve at \( u = 0.8 \) into two sub-curves.

Select **Techniques** followed by **Curve Subdivision**. The result is shown in the left figure below. The two curve segments are shown in black and yellow. Note that your system may use different colors. As mentioned earlier, the second segment is always the current curve. As the \( u \)-indicator moves to \([0.8,1]\), the sub-interval of the vertical slider is shown in the same color as that of the curve. Keep in mind that the current curve is always thicker and shown in brighter color. The right figure below is the partition of unity of this curve segment. Please note that the domain is \([0.8,1]\).

If you slide the \( u \)-indicator down to a location less than 0.8, the current curve will automatically switch to the first curve segment, in this case, the black one. The domain of the current curve on the vertical slider is also shown in black, and the partition of unity window will show that of the current curve.
A subdivided curve can be further subdivided. For example, if the curve segment defined on \([0,0.8]\) is further subdivided at \(u = 0.3\), we have the left figure below. The black curve segment is the one on \([0,0.3]\), while the blue one is the curve segment on \([0.3,0.8]\). Since this subdivision has no impact on the curve segment defined on \([0.8,1]\), the yellow curve is the same as in the above figure.

Maintaining \(C^1\)-Continuity

After curve subdivision, moving control points becomes an important issue. In the following figure, the joining control point is marked with a blue rectangle. What if it is moved to a new location? If its two adjacent control points, one for each curve segment, are not moved accordingly, the two segments will create a wedge at the common point (i.e., the tangent vectors of the curve segments at the common control point are different). Note that moving one of its adjacent control point would give us the same effect. To overcome this problem, this system also maintain the relationship that both curve segments are tangent to the line segment containing the common control point. When the common control point or one of its adjacent control point is moved, necessary action will be taken to guarantee this "tangency" condition (i.e., \(C^1\)-continuity).
Let us move the common control point, the one marked with a blue rectangle, to a new place. The result is shown in the left figure below. As you can see, this system moves the common control point as well as its two adjacent neighbors. Therefore, the curve segments are still tangent to the line segment at the common control point.

Let us now move one of the two neighbors of the common control point, say the one marked with a blue rectangle in the left figure above. Once that control point is moved, the control point at the other end will also move. In fact, these two control points will rotate about the common control points to keep these three control points collinear. Moreover, the length between this endpoint and the common point will be adjusted to maintain $C^1$-continuity. A result is shown in the right figure above.
The Display & Tracing Options Window provides a way to set various display and tracing options. As shown below, the window is divided into four parts. The first part is to indicate if the options should affect all curves or the current curve. The second part contains 11 display options. The third part provides a short cut for selecting the next curve or the previous curve to be the current curve. Finally, the four part is for closing this window. Clicking on these buttons activate or deactivate the select options. Options selected will be marked with a X. Some buttons are indented. This means those options are activated only if their "parent" options have been activated.

If you want the selected options applied to all curves on the drawing canvas, click on All Curves. Otherwise, the options will be applied to the current curve.

The display options include the following:

- **Control Points**
  If this option is activated, control points will be displayed. If w/ index is also activated, the control point indices (or IDs) will be shown.

- **Tracing Point**
  If this button is activated, the tracing point will be displayed. As the little triangle (i.e., the u-indicator) in the right-hand side of the vertical slider moves, its corresponding point, the tracing point, on the curve moves accordingly.

- **Control Polygon**
  If this button is activated, the control polygon will be shown.

- **Convex Hull**
  If this button is activated, the convex hull of the current u will be shown.

- **de Casteljau's Alg./de Boor's Alg.**
  When the current curve is a Bézier curve, this button shows de Casteljau's Alg.; otherwise, it shows de Boor's Alg.. If this button is activated, the de Casteljau or de Boor net will be shown. If w/ points is also activated, the endpoints of each segment will also be shown. See Tracing the Curve - de Casteljau's and de Boor's Algorithm for further details.

- **Tangent, Binormal & Normal**
  If this button is activated, the tangent vector, binormal vector and the normal vector at the tracing point (i.e., the moving triad) will be shown. If button Curvature Sphere is also activated, the curvature sphere at the tracing point will also be shown. Moreover, if Roller Coaster is activated, the Roller Coaster Window appears showing the roller coaster effect. Note that the curvature sphere and the roller coaster effect are independent of each other. See The Moving Triad, The Curvature Sphere, and The Roller Coaster Effect for further details about the moving triad, curvature sphere and roller coaster effect, respectively.

- **Coordinate Axes**
  If this button is activated, the coordinate axes will be shown in the drawing canvas.

Clicking Next Curve and Pre Curve selects the next curve and previous curve to be the current curve. If the current curve is the last (resp., first) curve, the next (resp., previous) curve is the first (resp., last) curve.
Finally, clicking on Dismiss closes this window.
The Partition of Unity Window displays the partition of unity of a curve and its basis functions. The left figure below shows a clamped B-spline curve of degree 4. The right figure below is its partition of unity window. The small triangles show the position of knots. As \( u \) changes, a vertical bar also moves in this partition of unity window. The values of basis functions are stacked up as shown with fixed vertical bar in the right side of the window. Different basis functions are shown in different colors. Thus, the fixed vertical bar is "partitioned" by different color intervals and this is the way of partitioning \([0,1]\) at \( u \). Note that as \( u \) changes, the partition changes as well.

If the knot vector is modified, the partition of unity changes on-the-fly, reflecting the change being made with the knot vector. The following figures show the result of merging some knots into multiple knots. The new knot vector has the same end knots, but the internal knots are 0.25(2), 0.5(3) and 0.75(3). That is, some of the knots are combined to create multiple knots: 0.25 with multiplicity 2 and 0.5 and 0.75 with multiplicity 3. Please note the change of knot positions in the partition of unity window.

Please also note that before merging knots a basis function is non-zero on five knot spans. After merging knots, some basis functions are non-zero on only one knot span. In general, none of the basis functions is non-zero on more than two knot spans.
The **Triangular Computing Scheme Window** is used to show the triangular computing scheme of de Casteljau's and de Boor's algorithms. This window has three buttons. Button **STEP** allows a user to see the computation column-by-column, button **RESET** resets the current computation and starts over, and button **DISMISS** closes this window.

The following figures show the final stage of the computation of de Casteljau's algorithm for a Bézier curve of degree 5. Note that all control points of a Bézier curve are listed on the first column, while the point on the curve is shown on the right-most column. Note also that for a B-spline or NURBS curve, not all control points are involved in the computation.

See **Tracing the Curve - de Casteljau's and de Boor's Algorithms** for the details.
The Hodograph (First Derivative) Window and Second Derivative Window

The hodograph of a curve is actually its first derivative. The hodograph (i.e., first derivative) of a Bézier (resp., B-spline) curve of degree \( p \) is a Bézier (resp., B-spline) curve of degree \( p-1 \) whose control points can be computed easily from the given control points. As a result, computing the second derivative of a Bézier or a B-spline curve is a simple task, because one can construct the control points of the hodograph and then construct the hodograph of the hodograph. In fact, it is not difficult to fine a formula for higher order derivatives.

Selecting **Hodograph (First Derivative) and Second Derivative** will open the hodograph and second derivative windows, respectively. Select **Dismiss** to close the window. These windows have two more buttons. **Magnify** and **Shrink** can be used to zoom in and zoom out with respect to the coordinate origin. If a curve has many control points, it is possible that some control points in the hodograph and second derivative windows are too populated to be shown clearly. Thus, **Magnify** and **Shrink** are useful in this and other cases.

In the following, the left figure is a Bézier curve of degree 4 with \( u = 0.42 \). The middle one is the hodograph window and the right one is the second derivative window. In all windows, the control points are clearly shown. As \( u \) moves on the given curve \( p(u) \), the points on \( p'(u) \) and \( p''(u) \) are shown on-the-fly. The tangent (resp., second derivative) vector is the vector from the coordinate origin to \( p'(u) \) (resp., \( p''(u) \)).

However, rational Bézier and NURBS curves do not have such an elegant property. Their hodograph and second derivative curves are no more rational Bézier and NURBS curves and as a result no control points are shown in the hodograph and second derivative windows.

The above shows the hodograph and second derivative of a rational Bézier curve. In this case, **Shrink** is needed to bring the hodograph and second derivative curves into the window area.

Further discussion can be found in **The First Derivative (Hodograph) and Second Derivative.**
The Change Color Scheme Window is for changing the colors being used for most displayable objects on the drawing canvas. A detailed discussion can be found in Changing the Color Scheme.
The Profile Instances Control Window

This window is not for a typical curve design session. It is only used in an experimental component for designing swept surfaces using skinning. See Using Skinned Surfaces to Design Swept Surfaces. Since it is experimental and is not stable yet, this window may be changed or even dropped in future version of DesignMentor.

For the details of swept and skinned surfaces, please refer to Cross Section Design, Simple Swept Surfaces and Skinned Surfaces.
Menu Structure: File

The File button consists of the following five items:

- **New Scene**
  This item is for clearing the draw canvas. It is normally used when starting a new design.

- **Import Scene**
  This is used to load curves from a file *without* clearing the drawing canvas. Therefore, the drawing canvas will contain *both* existing curves and the new curves read from the indicated file.

- **Load Scene**
  Clear the draw canvas and load curves from the indicated file.

- **Save Scene**
  Save the curves on the drawing canvas to the indicated file.

- **Save Color Scheme**
  Save the colors of the displayable objects to the file `curve.rc`.

- **Exit**
  Exit this program.

Details of **Import/Load/Save Scene** can be found in *Saving and Loading Your Work*, and **Save Color Scheme** in *Changing the Color Scheme*. 

This menu item is for specifying and displaying curve segments. The supported curve types are Bézier, rational Bézier, B-spline and NURBS curves, where different curve types have slightly different options. Once a curve type is selected, its type is displayed in this item.

- **New Curve Segment**
  This item is for specifying the curve type. One of the following four can be selected:
  - **Bézier**
    - Bézier curves
  - **Rational Bézier**
    - Rational Bézier curves
  - **B-spline**
    - B-spline curves
  - **NURBS**
    - Non-Uniform Rational B-spline curves

Normally, one should select a curve type followed by creating control points. If control points have already been created on the drawing canvas with a wrong curve type, one can use Change Current Curve to to change the curve type without changing the control points.

Once B-spline or NURBS is selected, this program assumes the curve has degree 3 and displays this degree in the Degree menu. If degree 3 is not suitable for a particular application, it can be changed with the Degree menu. Note that the maximum degree is 10 for B-spline and NURBS curves, which is sufficient for most applications.

- **Show Curve Segment**
  If the curve type is Bézier or rational Bézier, this item is unnecessary since the curve will be generated on-the-fly as control points are being created.

For a B-spline and NURBS curve, it is required to tell this system if the curve should be displayed as an open curve, a clamped curve or a closed curve. These three types of curves correspond to with Uniform Knots, with Clamped Knots and with Closed Knots, respectively.

The above figures, from left to right, show open, clamped and closed B-spline curves of degree 4 with the same set of control points. An open B-spline/NURBS curve does not pass though the first and the last control points. This system displays an open B-spline/NURBS curve with uniformly spaced knots. A clamped B-spline/NURBS curve is tangent to the first and last legs at the first and last control points, respectively. This system displays a B-spline/NURBS curve with the first \( p+1 \) knots and the last \( p+1 \) knots clamped to 0 and 1, respectively. All internal knots are uniformly spaced. For a closed B-spline/NURBS curve, the knots are still uniformly spaced; but some knots at both ends are fixed and cannot be modified.

Note that except for closed B-spline/NURBS curves, one can always manipulate the knot vector to make an open B-spline/NURBS a clamped one and vice versa.

This item can be selected at any time to change its knot pattern after a B-spline and NURBS curves has been displayed.

- **Change Current Curve to**
  This item allows a curve on the drawing canvas to be changed to another type using the same set of control points. There are four choices under this item: Bézier, Rational Bézier, B-spline and NURBS. Once one of them is selected, the curve is "converted" to the desired type. Note that this conversion simply "forgets" the original curve type and uses the same set of control points to generate a new one. As a result, when B-spline or NURBS are selected, its degree and its knot pattern (i.e., open, clamped and closed) are required before a curve can be displayed.

- **Next Curve Segment**
  Make the next curve the current curve segment. If the current curve segment is the last one, the next curve segment is the
**Previous Curve Segment**

Make the previous curve the current curve segment. If the current curve segment is the first one, the previous curve segment is the last curve.

---

**Knot Vector**

The left side of the vertical slider in the drawing canvas is used to display the knot vector. The following figures, from left to right, show the knot vectors of an open, a clamped and a closed B-spline curve of degree 4. In the display, a knot is shown as $p(q)$, where $q$ is the knot value and $p$ is its multiplicity. For example, in the knot vector of the open B-spline curve, $1(0.760)$ means 0.760 is a knot whose multiplicity is 1 (i.e., 0.760 is a simple knot). In the knot vector of the clamped B-spline curve, the clamped knots are $4(0.000)$ and $4(1.000)$ which means knots 0 and 1 are of multiplicity 4. For a closed B-spline and NURBS curve, several knots near 0 and 1 are shown in a different color, which means they cannot be modified (i.e., fixed to make the curve closed).
There are four different edit modes, (1) the **insert** mode, (2) the **select** mode, (3) the **move** mode and (4) the **delete** mode. The insert mode pointer is a cross, the select mode and move mode pointer is a north-west pointing arrow, and the delete mode pointer is a skull. The insert mode is for inserting control points; the select mode is for selecting a control point for further processing; the move mode is for moving control points; and the delete mode is for deleting control points. Note that clicking a control point in the move mode also selects that control point.

The **Edit** button consists of the following six items:

- **Create**
  This puts the system to the insert mode. The mouse pointer becomes a cross. Clicking in the drawing canvas creates control points. The most recent created control point is shown in a color that is different from the color for displaying other control points. All control points are represented with squares with numbers attached, the control point numbers. After selecting Create, the program will be in the insert mode until one of the non-insert edit items is selected (i.e., Select, Move and Delete).

- **Insert After**
  After a control point is selected with Select or click at that control point in the move mode, choosing Insert After allows to insert new control points after the selected one by clicking in the drawing canvas. All original control points after the selected one will receive new control point numbers.

The above left figure has 4 control points with \( p_1 \) being selected (i.e., shown in green). If Insert After is selected and then click at a position on the drawing canvas, a new control point \( p_2 \) is added and the original \( p_2 \) and \( p_3 \) are renumbered as \( p_3 \) and \( p_4 \).

Insert After will put the system to the insert mode until one of the non-insert edit modes is selected (i.e., Select, Move and Delete). As a result, one can add several control points. In fact, Insert is a special case of Insert After.

- **Insert Before**
  Insert Before is similar to Insert After except that the new control point is inserted before the selected one. The new control point takes the control point number of the selected one and the selected control point and all consequent ones will be renumbered.

In the above left figure, the first control point \( p_1 \) is selected. If Insert Before is selected and then click at a position on the drawing canvas, a new control point \( p_1 \) is added and the original \( p_1, p_2 \) and \( p_3 \) are numbered as \( p_2, p_3 \) and \( p_4 \).

- **Select**
  Selecting Select will put the system to the select mode (i.e., the mouse pointer is a north-west pointing arrow). Clicking a control point selects it for other processing.

- **Move**
**Move** is for putting the program in the *move* mode so that control points can be moved. One can drag control points to new locations. In general, **Move** is used right after all control points are created so that the positions of control points can be further adjusted to obtained the desired curve shape.

- **Delete**
  
  **Delete** is for deleting control points. When **Delete** is selected, the pointer becomes a skull. Clicking a control point removes it. For Bézier and rational Bézier curves, deleting a control point causes redrawing the curve. However, for B-spline and NURBS curves, since deleting a control point invalidates the fundamental equality $m = n + p + 1$, where $m+1$, $n+1$ and $p$ are the number of knots, number of control points and degree, the corresponding curve will not be displayed until **Show Curve Segment** is selected.

Further information can be found on page **Modifying Control Points**
This menu item provides you with a few advanced features of this system. These include knot insertion, degree elevation and curve subdivision. All of these options apply to the current curve.

- **Degree Elevation**
  This will increase the degree of the current curve by one without changing the shape of curve. See [Degree Elevation](#) for the details.

- **Curve Subdivision**
  This will subdivide the current curve at a specific value of $u$. See [Curve Subdivision](#) for the details.

- **Knot Insertion**
  This will insert a new knot into the knot vector of the current curve without changing the shape of the curve. See [Knot Insertion](#) for the details.

- **Knot Splitting**
  This will split a multiple knot into two, one being a simple knot while the other being the original knot with multiplicity decreased by one. Note that this option will change the shape of the current curve slightly. See [Knot Splitting](#) for the details.

The next five menu items are used with [Cross Section Design](#) for generating ruled surfaces, surfaces of revolution, swung surfaces, swept surfaces and skinned surfaces. See [A User Guide to the Surface Subsystem of DesignMentor](#) for the details.
The **Degree** menu item is for selecting the degree of a B-spline or a NURBS curve. If the selected curve type is Bézier or rational Bézier, **Degree** has no effect since the degree is determined by the number of control points. The degree of this curve is shown in this menu item.

Before a B-spline or a NURBS curve can be displayed, its degree (from 1 to 10) must be selected with **Degree**. The minimum degree and maximum degree of a B-spline or a NURBS curve is 1 and 10, respectively. The selected degree is shown in this menu item.
The **Window** button provides the users with additional information. It includes the following items: **Display & Tracing Options**, **Partition of Unity**, **Triangular Computing Scheme**, **Hodograph (First Derivative)**, **Second Derivative**, **Change Color Scheme** and **Profile Instances Control**. Selecting one of these items opens the corresponding window.

- **Display & Tracing Options**
  This will bring up the **Display & Tracing Options Window** for various display options. See [The Display & Tracing Options Window](#) for the details.

- **Partition of Unity**
  This will bring up the **Partition of Unity Window**, which displays the partition of unity by the basis functions at the current value of $u$. See [The Partition of Unity Window](#) for the details.

- **Triangular Computing Scheme**
  This will bring up the **Triangular Computing Scheme Window**, which provides a detailed step-by-step computation of de Casteljau's algorithm if the current curve is a Bézier curve. Otherwise, it displays the stepwise computation of de Boor's algorithm. A detailed discussion is in [Tracing the Curve - de Casteljau's and de Boor's algorithms](#).

- **Hodograph (First Derivative) and Second Derivative**
  This will bring up the **Hodograph (First Derivative) Window** and **Second Derivative Window**, respectively. The use and meaning of these two windows can be found in [The Hodograph (First Derivative) Window and Second Derivative Window](#) and [The First Derivative (Hodograph) and Second Derivative Curves](#).

- **Change Color Scheme**
  This window is for changing the colors used to display various objects. See [Changing the Color Scheme](#) for the details.

- **Profile Instances Control**
  This window is only useful in **Cross Section Design** and is in a very experimental stage. It may disappear or be modified in future versions. See [The Profile Instances Control Window](#) for additional information and [Using Skinned Surfaces to Design Swept Surfaces](#) for its use.
Menu Structure: Help

There are two items under menu **Help**: The **About...** window and the **Version** window. The former tells you the program name of this system, its version and creation date, the programmer(s) and the platform and other software tools used. The latter shows you the information, including a disclaimer, about this product.