Chapter 7

Drawing

7.1 Why Draw?

Artists draw, architects draw, graphic designers draw, cartoonists draw, but scientists and engineers mostly plot. There are several circumstances in which scientists are forced to draw, too, including:

- To make schematic diagrams.
- To annotate or elaborate the graphs plotted by Matlab or other plotting software.
- To trace part or all of a scanned image.

Many articles contain schematic diagrams: the layout or design of experimental apparatus, a computer program flow chart, a graph of boxes and arrows to indicate the relationships between concepts. The sine function can be graphed in Matlab because it is described by a mathematical formula, but what algorithm describes the structure of a mass spectrometer? The crucial reason that drawing is needed is that: Mathematics can be PLOTTED but concepts must be DRAWN.

Plotting software has shown a steady increase in the range of options for annotating a graph. Matlab, for example, has only recently added the capability to interpret strings as Latex code, thereby making it easy to add Greek letters, subscripts and complete mathematical expressions to graphs. Nevertheless, in the continuing "arms race" between technology and the aspirations and inventiveness of scientists and engineers, it still seems to be a common occurrence to want more than the plotting software is presently able to provide. Drawing programs like Adobe Illustrator, since their whole purpose is to create dazzling curves, text and shapes, always offer bells-and-whistles far beyond those of plotting software. A good strategy is to plot the graph as elaborately as possible within Matlab (or other plotting software) and then append additional labels, legends and shapes within a drawing program.

A crucial technical point is that many graphical formats which Matlab supports as output-to-a-file are also supported as import-from-a-file by illustration software. An equally crucial conceptual point is that many of the options available within a graph library are really drawing rather than plotting capabilities.

Scanners are a great boon to technical graphics because many journal papers contain one or more figures borrowed from previous work. These "history" graphs provide a context for the new work to come. Unfortunately, "as-is" reproduction is often unsatisfactory. The original source may have used different notation and different scales

and have displayed five variables where only one is needed for the introduction of the current work.

The pre-scanner strategy for altering a previously-published figure was to have a xerox traced and redrawn by a professional draftsman. Unfortunately, such artists have become an endangered species. Furthermore, the scientist is still responsible for making the technical decisions: what to plot, where to plot it, what and how to label.

The modern strategy is "scan-and-draw". A scanner is a xerox machine that uses the laser trace of the document placed on its glass plate to write a computer file instead of to attach a spot of ink to paper. The scanned image can be reprinted "as-is". If it needs modification, these must be performed using a drawing program.

The scanning software that controls the image capture always incorporates a limited suite of drawing tools. The scanned image must be cropped, usually rotated a little to compensate for twisting of the original, and stray marks erased. If notation needs to be changed or the labels altered, then this demands a second stage of processing through a program like Adobe Illustrator.

7.2 Illustrating with TeX and Postscript

The book by Goosens, Rahtz and Mittelbach(1997) describes several wyas to create illustrations directly in Latex. One strategy is to use Postscript commands. This is satisfactory for drawing simple polygons, arcs, and schematics. A second strategy is to use a special-purpose package. TeX and LaTeX afficionados have developed sets of macro commands which are catalogued in Goosens *et al.*(1997). Specialized packages include:

- Feynman diagrams (particle physics) [FeynMF]
- chemical [XYMTEX]
- music [MusiXTEX, abc2mtex, MPP, midi2tex]
- chess [chess]
- other games including Go, Xiangqi (Chinese chess), backgammon, bridge and crosswords

The package **X**Y-**pic** is particularly powerful. It can be used to create complicated diagrams with arrows, matrix-like structures connected by arrows, finite-state diagrams, directed graphs and flow charts, circuit diagrams, two-cell diagrams, arcs, circles and ellipses.

The special purpose packages may be the best option for specialized applications. However, they are much more limited than general purpose drawing packages. As we shall explain in the rest of the chapter, it is fairly easy to generate a standard Feynman bubble diagram or wavy line either in a drawing program or as the output of a Matlab, Maple or Mathematica function. This standard element is easily resized, rotated and translated either in drawing software or scientific software. One can easily build up a rather complex diagram out of a handful of standard elements.

7.3 Drawing in Matlab

7.3.1 Introduction

Matlab, although not designed to be a drawing program *per se*, is nevertheless capable of creating very complicated images. The disadvantage of drawing in Matlab is that it lacks the convenient, mouse-driven drawing tools of a program like Adobe Illustrator. The strength of Matlab is the power to draw-by-algorithm. If one needs to create an image with a complicated shape but knows a mathematical formulation for computing the points of the shape, then Matlab may be greatly superior to an illustration program.

Fig. 7.1 is a sampler of drawings created entirely in Matlab. Each insect and each child was drawn by repeatedly calling a single function with arguments that specified the position and size of each figure. The functions were borrowed from Nakamura (1996) and no doubt required a good deal of debugging and fiddling to get the shapes just right. It is almost always faster to draw a single insect or stick-person with an illustrator program than with Matlab. However, for "volume production", Matlab is just as fast or faster. One can duplicate and move copies of a subfigure in an illustration program, but it is just as quick to invoke a function with a translated argument in Matlab.

One could draw an assortment of circles of random diameters at random locations using the circle-drawing tool of an illustration program, but each circle would then have to be drawn individually. When the copies are not identical, manually rescaling and moving them in an illustration program becomes quite tedious.Furthermore, Matlab's **rand** function ensures that the random sizes and positions are *really* random — or at least really pseudo-random.

In a similar spirit, the plane was created using a standard wing shape defined by a mathematical algorithm. The same wing section was employed, with rescaling and translation, for both halves of the main wing, both tailplanes, and the rudder.

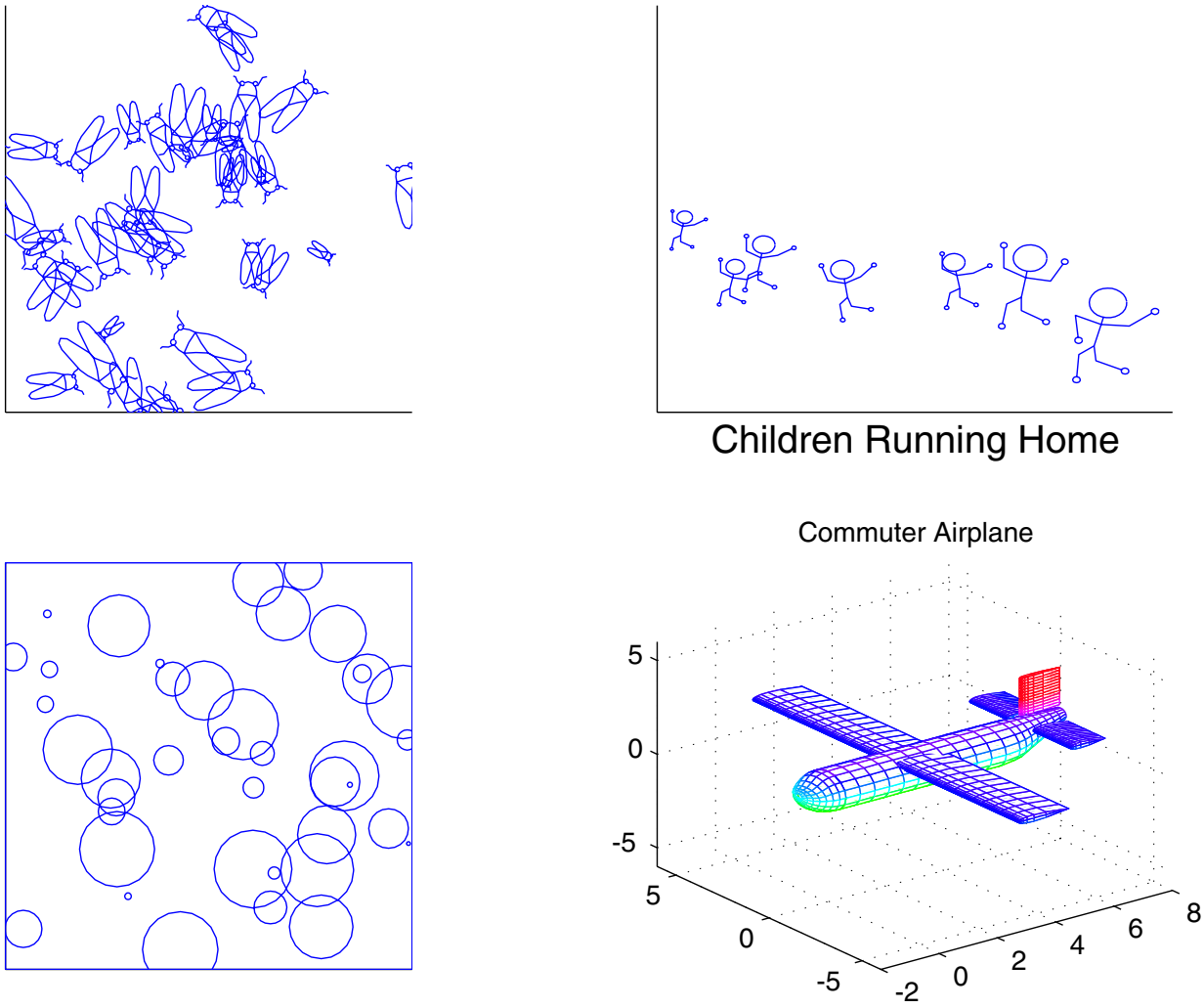


Figure 7.1: A potpourri of drawings in Matlab, taken from the book by Nakamura(1996)
.

7.3.2 Importing Images Into Matlab

Matlab can import a variety of graphic formats and display them using its **image** family of commands. After invoking **hold on**, one can then modify these pictures by overlaying additional graphs and text. Finally, the modified picture can be printed or saved to a file.

The most important limitation on import is that although Matlab can *save* figures as Encapsulated Postscript (EPS) and, on the Macintosh, PICT files, it unfortunately cannot *import* either of these formats. However, Matlab can directly import other popular formats such as TIFF and JPEG. Adobe Illustrator can import EPS files and save them as TIFF or JPEG; specialized graphic converters such as the shareware GraphicConverter (on the Macintosh) will also do the necessary file conversion. Once one has a file in an importable format, the statements

```
[Simage Smap]=imread('SailingShiponReef','tiff');
xa=linspace(-1,1,30);    ya=linspace(-1,1,40);
imagehandle=image(xa,ya,Simage);
```

will read and display an image. The first argument of **imread** is the file name (with the termination .fmt omitted) and the second argument is the file format. The **linespace** commands (not part of the **image** family) are used to create vectors to define the horizontal and vertical axes. The **image** command actually uses only the first and last elements of the vectors so that **imagehandle=image([-1 1],[-1 1],Simage);** would actually produce the same image with the same axis scales and tick marks.

7.3.3 Tracing an Image in Matlab

The command **ginput** puts a cross-hair cursor on the screen, and then stores the x and y values of the location of the mouse each time it is clicked until the function is terminated by hitting the Return key. The syntax is

```
[x,y]=ginput
```

This command is very powerful because it allows one to easily *trace* parts of images. The arrays **x** and **y** can then be transferred by a copy-and-paste operator to a function subprogram which will then be able to reproduce the traced curve every time that it is called.

7.3.4 Translating, Rescaling and Rotating

To exploit a function that draws an image effectively, we need to be able to translate, rescale (or "dilate") and rotate the image. This is rather easy to do. We can build the necessary properties into a standard image-drawing template. We can then retitle the template and pop in vectors that define the image for some standard size and orientation, and away we go!

For example, a two-dimensional template is given by

```
function linehandles = Drawtemplate2D(x0,y0,dilation,rotation
hold on

% Part I: Define arrays x, y which contain the points for
% the figure with center at x=0,y=0,
% dilation=1 and rotation angle=0

% Translate, dilate and rotate arrays x,y
xxxx=x0*ones(1,length(x))
        + dilation*(x*cos(rotation) - y*sin(rotation));
yyyy=y0*ones(1,length(x))
        + dilation*(x*sin(rotation) + y*cos(rotation));

handle1=plot(xxxx,yyyy,'k-');   % plot arrays xxx, yyy
hold off
linehandles=handle1;
```

The first step in the function is to generate an array of points for the figure in canonical form: a certain standard position, orientation and size. For example, to generate a square with sides normalized to a length of two and sides parallel to the x and y axes:

x=[-1 1 1 -1 -1]; y=[-1 -1 1 1 -1];

The second step is to translate, dilate and rotate the arrays of points. The rotation is performed by multiplying each (x, y) pair of points by a two-by-two rotation matrix whose elements are cosines and sines of the rotation angle:

xx=x*cos(rotation) - y*sin(rotation);
yy= x*sin(rotation) + y*cos(rotation);

"Dilation" means resizing the figure in all its dimensions by multiplying by a constant $dilation$:

xxx= dilation*xx; yyy= dilation*yy;

The transformation which translates the figure from a center at (0, 0) to a figure whose center is at (x_0, y_0) is

xx = x + x0, yy = y + y0

The last line of the template ensures that the function returns a handle to lines graphed by the **plot** command. One can then further manipulate the object in the main program (called a "script") in Matlab or directly from the command line. The syntax is

set(linehandle,'Color','g','LineWidth',2,'LineStyle',':')

to change the color of all lines in the function's figure to green, alter the width to 2 pts (where one point = (1/72) of an inch) and change the lines from solid to dotted.

As an example of two-dimensional figure drawing, the complete code to compute an ellipse of arbitrary size, position and orientation and arbitrary semimajor and semiminor axes is given in Matlab by

```
function thandle=  ellipB(x0,y0,dilation,rotation,rx,ry)
% x0, y0:  coordinates of center point
% rx, ry:  radii in x and y  directions

% Part I: Compute ellipse with axes parallel to
% x and y axes and center at (0, 0)
delt = 2*pi/30;     t = 0:delt:2*pi;
x=rx*cos(t);   y = ry*sin(t);
% Translate, dilate and rotate the ellipse
xxxx =x0 + dilation*(cos(rotation)*x - sin(rotation)*y);
yyyy =y0 + dilation*(sin(rotation)*x + cos(rotation)*y);
thandle=plot(xxxx,yyyy,'k-');
```

The ellipse routine takes two extra arguments, omitted from the template, to specify the radii of the ellipse, which are equal to half the axes. It will often be necessary to add extra arguments that are specific to a particular species of figure. However, the translation, dilation and rotation lines are exactly as in the template.

Fig. 7.2 illustrates the result. The color, line width, and linestyle was changed by using the **set** command in the calling script. The filled ellipse was produced by creating a new function, **ellipBfill**, which added **fill(xxxx,yyyy,colorstring)** just after the plot statement, and appended one new argument, the colorstring. One could have combined these two functions into a single function with a variable number of arguments, but it seemed less confusing to have two different functions.

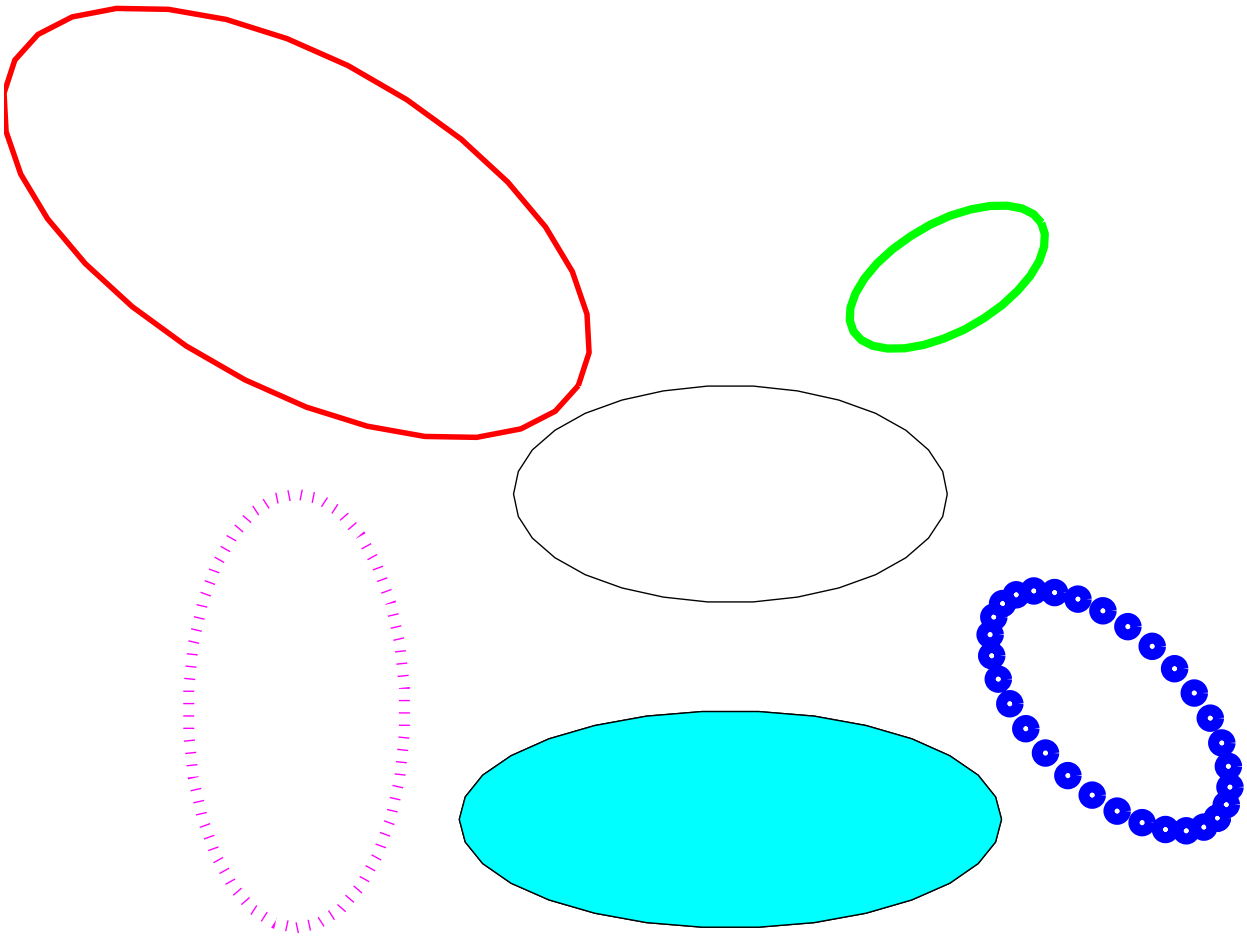


Figure 7.2: An assortment of ellipses of various rotation angles and sizes (dilations). The colors and linestyles were altered in the calling script using the **set** command, applied to the line handle returned by the ellipse-drawing function, **ellipB**.

A three-dimensional template is

```
function h=Drawtemplate3D(x0,y0,z0,dilation,rotx,roty,rotz)
% (x0,y0,z0) is center of the figure (Cartesian coordinates)
%    dilation = size, relative to the default size
%  rotx = rotation angle  about x-axis
%  roty = rotation angle  about y-axis
%  rotz = rotation (radians)  about z-axis
hold on
% FIRST BLOCK: define the points of the figure
%  as the arrays x,y,z in Cartesian coordinates.
       FUNCTION-SPECIFIC STUFF GOES HERE
       % Create the rotation matrices
xcos=cos(rotx);   xsin=sin(rotx);
ycos=cos(roty);   ysin=sin(roty);
zcos=cos(rotz);   zsin=sin(rotz);
          % rotation around the x-axis
xx=x;  yy=xcos*y - xsin*z;   zz=xsin*y + xcos*z;
         % rotation around the y-axis
yyy=yy;   xxx=ycos*xx + ysin*zz;   zzz=-ysin*xx + ycos*zz;
         % rotation around the z-axis
xxxx=zcos*xxx-zsin*yyy;  yyyy=zsin*xxx+zcos*yyy; zzzz=zzz;
           % Dilation
xxxx=dilation*xxxx;  yyyy=dilation*yyyy; zzzz=dilation*zzzz;
          % Translation
xxxx = xxxx + x0*ones(size(xxxx));
yyyy = yyyy + y0*ones(size(xxxx));
zzzz = zzzz + z0*ones(size(xxxx));
h=plot3(xxxx,yyyy,zzzz,'k-');    hold off
```

There are seven parameters: three to specify the location of the center of the figure $(x0, y0, z0)$, plus the size parameter *dilation* plus the three rotation angles. The dilation parameter specifies the size of the figure on the screen relative to the figure of standard size which is defined by the first block in the subroutine (where the arrays x, y, z must be computed). For example, if x, y, z specify a cube with unit sides, then $dilation = 3$ will graph a figure whose sides are all equal to three. Just as for the two-dimensional template, rotations are in *radians* in a *counterclockwise* direction.

The next listing shows an instance of this three-dimensional template, a routine for computing a cube. The canonical figure, computed by the first block of lines, is a cube

with unit sides. The centroid of the canonical figure is at the origin so that the vertices are at $x = \pm 1/2, y = \pm 1/2, z = \pm 1/2$. The line segments that define the sides are oriented so as to be parallel to the coordinate axes. The unit cube is by plotting each face as a square and then plotting four line segments to connect the two faces, vertex-to-vertex. Each face is specified by five points, even though a square face has only four vertices, because one vertex must be duplicated so that the plotted line segments will form a closed curve.

It is easy to generalize this listing to compute three-dimensional parallelpipeds, prisms and so on. However, when the sides are not all equal, one must add additional arguments to the seven parameters of the template to describe the shape of the canonical figure.

The listing could also be generalized by adding parameters to specify the color of the figure, linestyle, linetype and so on. Matlab's **vargin** allows one to call the function with a variable number of arguments so that optional parameters, such as color and linestyle, can be left out of the calling statement unless one wishes to change the defaults. For simplicity, the listing eschews such refinements, implicitly assuming that the color, linestyle and thickness will be changed by a **set** statement in the calling program.

Fig. 7.3 is a sample of the cubes produced by the listing. The circle of cubes was drawn by embedding the function in a **for ... end** loop such that the rotational angle about one axis varied through 2π and the translation $(x0, y0)$ was moved in a circle (with $z0 = 0$).

```
function handles=cube3B(x0,y0,z0,dilation,rotx,roty,rotz)
% (x0,y0,z0) is center of the figure (Cartesian coordinates)
%    dilation = length of sides of the cube.
%  rotx = rotation  about x-axis   roty = rotation,  y-axis
%  rotz = rotation (radians)  about z-axis,  hold on
% FIRST PART: Cube with UNIT SIDE and CENTER at x=0,y=0,z=0
x=[0 1 1 0 0] - 0.5*ones(1,5);   % Five points define one face
y=[0 0 1 1 0] - 0.5*ones(1,5);   z=[0.5 0.5 0.5 0.5 0.5];
x=[x x];    y=[y y];   z=[z -z];  % Second face
% Four parallel line segments to connect the faces
x=[x -0.5 -0.5  0.5 0.5 0.5 0.5 -0.5 -0.5];
y=[y -0.5 -0.5 -0.5 -0.5 0.5 0.5 0.5 0.5];
z=[z  -0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5];
       % Create the rotation matrices
xcos=cos(rotx);   xsin=sin(rotx);  ycos=cos(roty);
ysin=sin(roty);   zcos=cos(rotz);  zsin=sin(rotz);
           % rotation around the x-axis
xx=x;  yy=xcos*y - xsin*z;   zz=xsin*y + xcos*z;
          % rotation around the y-axis
yyy=yy;   xxx=ycos*xx + ysin*zz;   zzz=-ysin*xx + ycos*zz;
          % rotation around the z-axis
xxxx=zcos*xxx-zsin*yyy;  yyyy=zsin*xxx+zcos*yyy; zzzz=zzz;
            % Dilation
xxxx=dilation*xxxx;  yyyy=dilation*yyyy; zzzz=dilation*zzzz;
           % Translation
xxxx = xxxx + x0*ones(1,18);  yyyy = yyyy + y0*ones(1,18);
         zzzz = zzzz + z0*ones(1,18);
handles=plot3(xxxx(1:5),yyyy(1:5),zzzz(1:5),'k-',...
          xxxx(6:10),yyyy(6:10),zzzz(6:10),'k-',...
  xxxx(11:12),yyyy(11:12),zzzz(11:12),'k-',...
  xxxx(13:14),yyyy(13:14),zzzz(13:14),'k-',...
  xxxx(15:16),yyyy(15:16),zzzz(15:16),'k-',...
  xxxx(17:18),yyyy(17:18),zzzz(17:18),'k-'); hold off
```

7.4 Affine Transformations

The transformations illustrated in the Matlab code in the previous sections are special cases of so-called "affine transformations". For simplicity, the codes were restricted to translation, dilation, and rotation. It is easy to add the full range of affine transformations, however.

Definition 13 (Affine Transformation) *An affine transformation from old coordinates* (x, y) *to new coordinates* (x', y') *is one that can be expressed as a* MATRIX MULTIPLICATION *plus addition of a* VECTOR *(translation):*

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} x_0 \\ y_0 \end{vmatrix} \tag{7.1}$$

and similarly in three dimensions.

A number of special cases of affine transformations are worth noting:

1. Pure translation by (x_0, y_0):
   Matrix is zero; all points are translated by (x_0, y_0)

2. Plane Rotation through angle θ about the origin:
   $(x_0 = 0, y_0 = 0)$ and the matrix is
   $$\mathbf{R}_\theta \equiv \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix} \tag{7.2}$$

3. Plane Rotation through angle θ about a point (x_c, y_c):
   Denote the point (x, y) by the vector $\mathbf{p}$ and similarly $(x_c, y_c) \equiv \mathbf{p}_c$. Then
   $$\mathbf{p}' = \mathbf{R}_\theta(\mathbf{p} - \mathbf{p}_c) + \mathbf{p} \tag{7.3}$$
   where the rotation matrix is the same as defined in the previous item. In words, one translates from $\mathbf{p}_c$ to the origin, performs the rotation, and then translates back to the point $\mathbf{p}_c$.

4. Uniformation dilation: diagonal matrix with all diagonal elements equal to the scaling factor λ:
   $$\mathbf{D}_\lambda \equiv \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} \tag{7.4}$$

5. Nonuniform Dilation or Scaling: diagonal matrix with diagonal elements equal to the two different scaling factors λ, μ, respectively:
   $$\mathbf{D}_{\lambda,\mu} \equiv \begin{vmatrix} \lambda & 0 \\ 0 & \mu \end{vmatrix} \tag{7.5}$$

6. Inversion with Respect to the Original: diagonal matrix with diagonal elements equal to -1 ($x' = -x, y' = -y$):
   $$\mathbf{I} \equiv \begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix} \tag{7.6}$$
   Inversion preserves the orientation ("handedness") of a figure.

7. Reflection with Respect to the x-axis: diagonal matrix with diagonal elements equal to one and minus one ($x' = x, y' = -y$):

$$\mathbf{Re}^{(x)} \equiv \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \tag{7.7}$$

   Reflection always reverses orientation.

8. Reflection with Respect to the y-axis: diagonal matrix with diagonal elements equal to minus one and one ($x' = -x, y' = y$):

$$\mathbf{Re}^{(x)} \equiv \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} \tag{7.8}$$

9. Shear: x direction

$$\mathbf{S}_{\lambda}^{(x)} \equiv \begin{vmatrix} 1 & \lambda \\ 0 & 1 \end{vmatrix} \tag{7.9}$$

   In fluid mechanics, advection in the x-direction by a flow that varies only with y produces an x-shear. $y' = y$, so the "flow" does not alter the y coordinate. However, the x-coordinate is translated in the x-direction by an amount that depends on how far the point is from the origin, that is, by how large y is.

10. Shear: y direction

$$\mathbf{S}_{\lambda}^{(y)} \equiv \begin{vmatrix} 1 & 0 \\ \lambda & 1 \end{vmatrix} \tag{7.10}$$

11. Strain: x direction. Diagonal matrix with the top diagonal element equal to λ and the lower diagonal element equal to one.

$$\mathbf{T}_{\lambda}^{(x)} \equiv \begin{vmatrix} \lambda & 0 \\ 0 & 1 \end{vmatrix} \tag{7.11}$$

   This is a special case of a non-uniform dilation; shapes are distorted or "strained" because only the x-coordinate is stretched or compressed.

12. Strain: y direction. Diagonal matrix with the top diagonal element equal to one and the lower diagonal element equal to λ.

$$\mathbf{T}_{\lambda}^{(y)} \equiv \begin{vmatrix} 1 & 0 \\ 0 & \lambda \end{vmatrix} \tag{7.12}$$

   This is a special case of a non-uniform dilation; shapes are distorted or "strained" because only the y-coordinate is stretched or compressed.

If a figure or object is defined by a discrete set of points $(\vec{x}, \vec{y})$, the affine transform may be applied one-pair-of-points-at-a-time to transform the entire object.

The *general* affine transformation can be decomposed into the product (i. e., successive application) of elementary affine transformations in the list above.

Definition 14 (Isometry) *This is an affine transformation which is a "rigid-body" motion of a figure. Distances between all points of the object are preserved under an isometry. Any combination of translation, rotation and reflection is is an isometry; dilations, shears and strains are not isometries.*

Theorem 1 (Isometry) *There are four isometries in the plane:*

1. *translation*
2. *rotation*
3. *reflection*
4. *glide reflection*

The determinant of the 2×2 *transformation matrix is equal to* ± 1 *where orientation is preserved for the* $(+)$ *case and reversed when the determinant is* -1.

Definition 15 (Glide Reflection) *A "glide reflection" is a combination of translation with reflection with respect to a fixed line.*

Theorem 2 (Affine Transformation of Objects) *Under an affine transformation,*

- *Straight lines transform into straight lines.*
- *Conic sections transform into conic sections*
- *A curve of degree* n*, that is, a curve specified implicitly as the zero curve (algebraic variety) of a polynomial in* x *and* y *of degree* n*, is transformed into a curve of degree* n*.*

Circles, ellipses, parabolas and hyperbolas are all curves of degree 2, i. e., conic sections. Under an affine transformation, an ellipse can be transformed into a circle or vice-versa. However, a curve that opens to infinity such as a hyperbola cannot be transformed into a closed curve like an ellipse.

Geometry (and computer graphics) make heavy use of so-called "projective" transformations. Although we shall not explicitly discuss them here, the new coordinates (x', y') are *rational* functions of the old coordinates. Projective transformations can be easily incorporated into algorithmic drawing routines also. Indeed, invisible to the user, Matlab is performing projective transformations whenever the viewing angle is changed by the Matlab **view** command.

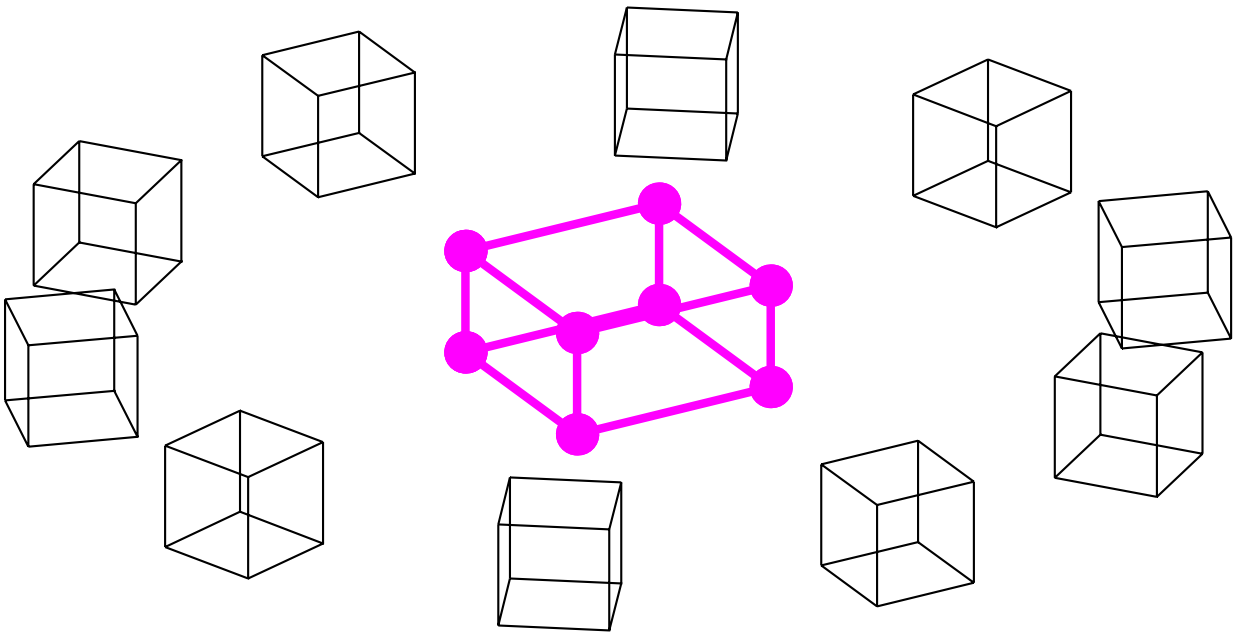


Figure 7.3: An assortment of three-dimensional cubes of various rotation angles and sizes (dilations) as computed by **cube3B**. The spheres at the vertices of the central cube were created by using **set(linehandles,'Marker','.','MarkerSize',30)** where "linehandles" is returned by the function **cube3B**.

7.5 Fractals: When the Algorithm is Mightier Than the Brush

Fractals, which are objects of complicated, self-similar structure with non-integral "capacity dimension", occur in many branches of science and engineering and art. To draw a tree or bush is very labor-intensive because a realistic image requires a large number of individual brush strokes or pencil lines.

In 1968, A. Lindenmayer introduced a family of algorithms, now known as "L-systems", to model the growth of living organisms, especially the branching of plants and trees. P. Prusinkiewicz later refined L-systems to a high art and published a book showing the great success of himself and others in generating very life-like plants and trees through L-systems rather than freehand drawing. Fig. 7.4 is an example.

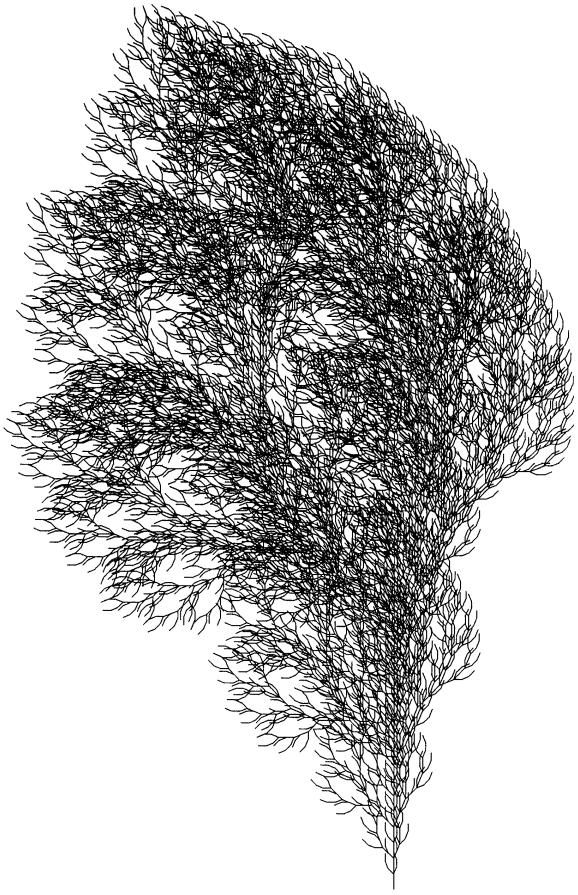


Figure 7.4: A bush-by-algorithm: a realistic-appearing bush generated by a Prusinkiewicz L-system.

Other objects of great complexity can be generated by very simple algorithms. The famous von Koch snowflake is constructed through an iteration in which the middle of each line segment is erased and replaced by two segments at an angle as illustrated in Fig. 7.5. The length of the curve increases without bound with each iteration, but the entire, infinitely long curve can always be enclosed within a rectangle of finite area.

Although artists have drawn beautiful trees for centuries without benefit of either mathematics or computers, it is hard to argue that for such complicated objects, the algorithm is the *faster* way. Drawing in Matlab is particularly appropriate for fractals.

For other kinds of graphical elements, it may be much faster to draw the object with a mouse or pen in an illustration program than in Matlab. With a pen, one can extend the curves to just the right length before clicking to fix the endpoint. In Matlab, the curves are specified as collections of points, and one may have to fiddle and fume for a long time before the points are just right.

Matlab is good for dodecahedrons and trees; Matlab is not good for faces.

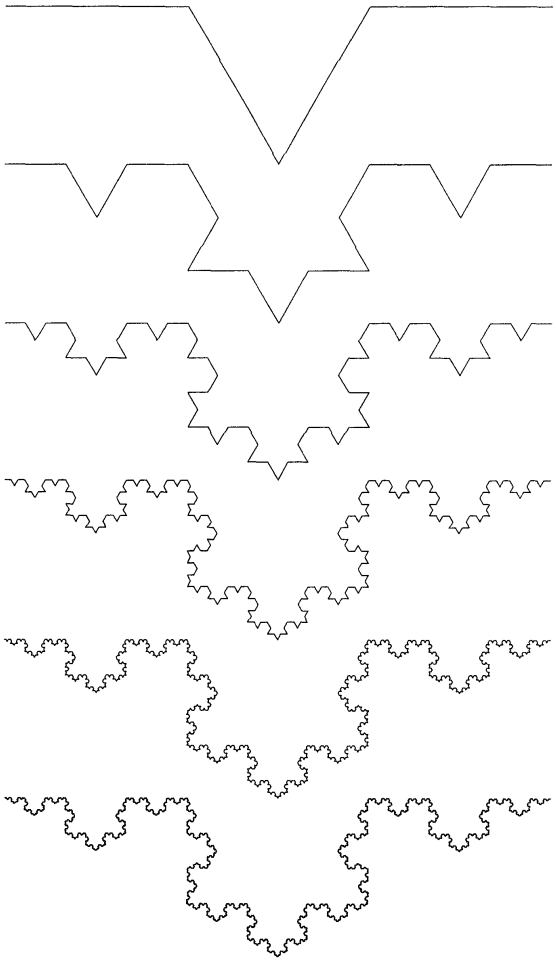


Figure 7.5: Six stages in the construction of the von Koch Snowflake. This curve, in the limit of an infinite number of stages, has an infinite length but encloses a bounded area.

7.6 Splines

In drafting, a "spline" is a thin strip of wood which is used to draw curves. Polynomial spline interpolation was developed by I. J. Schoenberg, an expatriate mathematician who spent the war years in a government laboratory where he saw draftsmen translating his calculations into graphs. In those pre-computer days, each graph was drawn from a very small number of plotted points: How were the gaps filled? To Schoenberg's amazement, the answer might be dubbed: Good Math from Carpentry. The draftsmen put pins into the paper at each of the calculated points, then snaked a thin strip of wood between the pins. The wood resisted bending: it spontaneously assumed the flattest shape consistent with the constraints of the pins. The draftsmen then interpolated between the computed points (pins) by tracing the shape of the spline.

A simpler procedure would be to connect each pair of points by a straight line. Such a piecewise linear graph would be quite unsatisfactory because, with only a small number of points, the graph would have very noticeable first derivative discontinuities.

In contrast, the shape of the spline was continuous in all its derivatives. A smooth appearance isn't enough, however; the curves need to be accurate. However, the springiness of the spline ensured that it curved only as much as necessary to interpolate (that is, go through) all the computed points. A little research and some conversations with engineers convinced Schoenberg that the spline minimized its curvature, subject to the interpolation constraints: it was the curve of smallest second derivative that intersected all the points.

Schoenberg was moved to pose a mathematical problem: Suppose one approximated a function $f(x)$ by a piecewise cubic polynomial, different for each interval between two interpolation points. The interpolation conditions at each endpoint of the interval give two conditions, but a cubic polynomial has four. Schoenberg demanded that the approximation minimize the curvature of the cubic polynomial. He showed that this condition was equivalent to the statement that both the first and second derivative of the piecewise cubic polynomial had to be continuous at the interpolation points even though different polynomials were used on either side of a given point. This gave enough conditions to uniquely determine the cubic polynomials for each interval of the entire domain.

Schoenberg's cubic-spline, often shortened to "c-spline" today, was a rousing success. It is a built-in option for Matlab's interpolation routines, and is included in every major library of numerical software. Unfortunately, splines have a drawback: the continuity conditions at the interpolation points imply that all the coefficients of the cubic polynomials must be found *simultaneously*; if even one point is altered, then the whole curve must be recomputed.

Splines are very powerful for "algorithmic drawing", as in Matlab. Many engineering design programs make heavy use of splines. However, drawing programs like Illustrator make use of a related but alternative strategy for drawing smooth curves between a limited number of fixed points.

7.7 Bezier Curves

7.7.1 Bernstein and Bezier: History

Around 1870, Karl Weierstrass proved an amazing theorem: any function $f(x)$, even if it was not continuous, could be approximated to arbitrarily high accuracy on an interval by a polynomial of sufficiently high degree. His proof was nonconstructive,

alas. Polynomial interpolation was known, but around the turn of the century, Karl Runge discovered that interpolation with evenly spaced points could diverge even for a function that was analytic and smooth everywhere on the target interval.

In 1912, Bernstein resolved the paradox by devising a noninterpolating polynomial approximation which was guaranteed to converge to any function for which Wierstrass theorem applies. Hurrah! Well, not quite. Unfortunately, Bernstein's approximation converges so slowly that the error at each point where f is continuous is asymptotically $C(x)/N$ where C is a function only of x and N is the degree of the approximating polynomial (Voronovsky's Theorem). Approximation theorists have been intrigued by Bernstein's polynomial ever since; among their amazing properties is that the Bernstein approximant of a convex function is always convex, and the approximant of a monotonic function is always monotonic. However, numerical analysts and engineers washed their hands of Bernstein's intriguing polynomials because of their very slow rate of convergence.

The situation changed in 1959 when the French car company Citroën hired a mathematician named Paul de Fajet de Casteljau to make a link because two-dimensional blueprints, then the standard result of engineering design, and Citroën's computer-

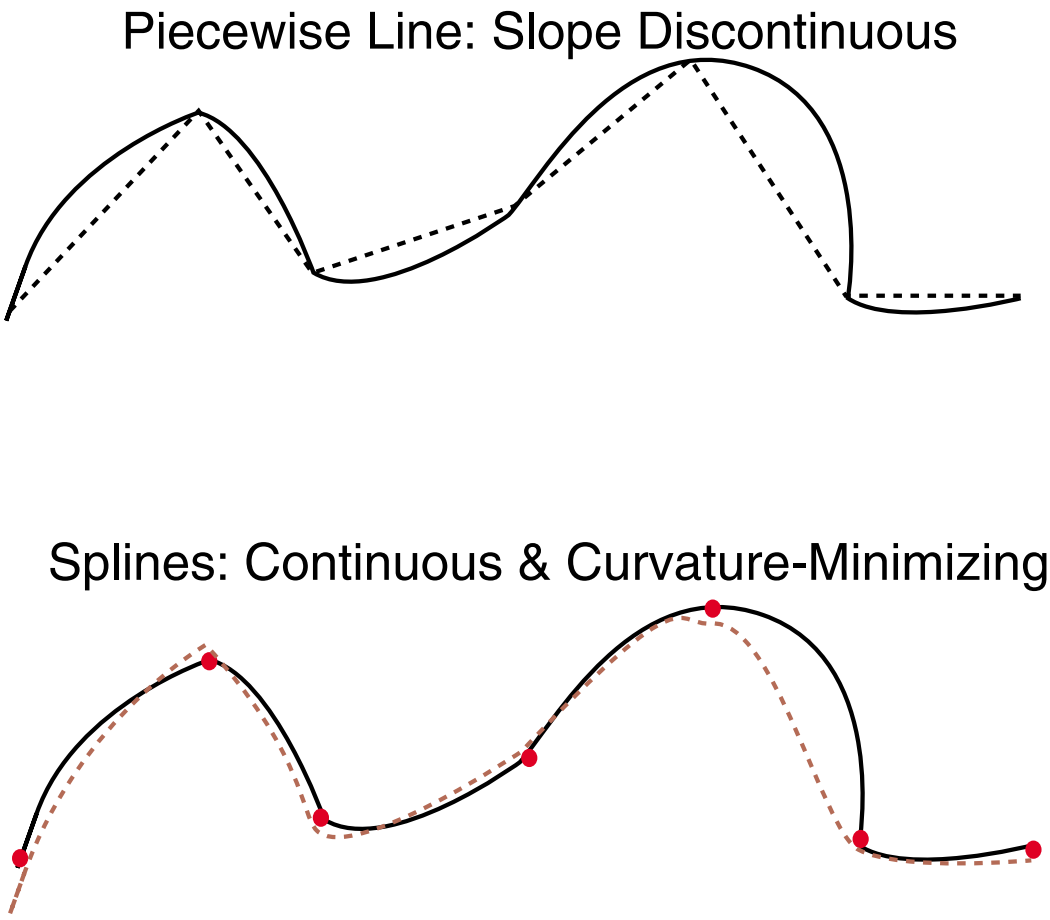


Figure 7.6: Top: piecewise linear approximation (dashed) of a function (solid). Bottom: spline interpolation where the dashed brown line is the wooden spline, and the curve traced around it, and the red disks are the interpolation points where pins are placed to constrain the spline.

controlled milling machines. He invented what are now called "Bézier curves". These are really just Bernstein approximants in a parameter t, $t \in [0,1]$, that parameterizes the curve.

His crucial conceptual breakthrough was to realize that high numerical accuracy was not important because the car designers had traced curves on the blueprints using rather inaccurate methods — artistic insight and curving sheets of plastic called "French curves". It was far important to feed a curve to the milling machine that was very *smooth* — that was what the engineers were trying to trace on the blueprints anyway. A wiggly curve would drive the milling machine nuts, increase cost, spoil aerodynamics, increase the difficulty of painting the car body, and generally be a VERY BAD THING. Because the Bernstein approximants are convex when the underlying curve is convex, and monotonic when monotonic, the Bernstein approximants never add spurious wiggles, but instead do a good job of smoothing the imperfections of curves traced on blueprints using mechanical drawing instruments.

Citroën tried to keep de Faget de Casteljau's discovery top-secret-crypto, which is why he never received full recognition. In time, though, a mechanical engineer at rival Rénault, Pierre Bézier learned enough about Citroën's system to recreate it. In contrast to Citroën, Rénault allowed Bézier to publicize his work widely. Bézier curves have dominated Computer-Aided Graphic Design (CAGD) for the last forty years.

In 1972, Gordon and Riesenfeld showed that Bézier curves were just a special case of B-splines, which allowed a unification among computer systems for CAGD. The Bézier curves have continued to dominate drawing programs although B-splines are more widely used for approximation when the function to be approximated is known accurately.

7.7.2 Bernstein Polynomials

"The Bernstein approximation of a continuous function lies between the extreme values of the function itself. ... Monotonic and convex functions yield monotonic and convex approximants. In a word, ..., the Bernstein approximants mimic the behavior of the function to a remarkable degree."

"[The convergence rate] is far slower than what can be achieved by other means. If f is bounded, then at a point where d^2f/dx^2 exists and does not vanish, $B_N(f;x)$ converges to $f(x)$ precisely like C/n. ... This fact seems to have precluded any numerical application of Bernstein polynomials from having been made. Perhaps they will find application when the properties of the approximant in the large are more important that the closeness of the approximation."
P. J. Davis(1975)[**?**], pg. 116 of his book.

As de Faget de Casteljau and Bézier recognized, engineering design and drawing are not problems in approximation of a function that is already completely and precisely defined. Rather, the goal is to create new smooth curves. The convexity and monotonicity properties of the Bernstein polynomials ensure that Bézier curves have very good "properties of the approximant in the large".

Definition 16 (Bernstein Polynomials) *The N-th Bernstein polynomial that approximates a function $f(x)$ is*

$$B_N(f;x) \equiv \sum_{j=0}^{N} f(j/N) \left(\begin{array}{c} N \\ j \end{array} \right) x^j (1-x)^{N-j} \tag{7.13}$$

or equivalently

$$B_N(f;x) \equiv \sum_{j=0}^{N} f_j \, \mathfrak{B}_j(x;N) \tag{7.14}$$

where f_j are the values of $f(x)$, the function being approximated, at a set of evenly spaced grid points on the interval $x \in [0,1]$ and where the $\mathfrak{B}_j(x;N)$ are the Bernstein basis functions. The binomial coefficient is

$$\left(\begin{array}{c} N \\ j \end{array} \right) = \frac{N!}{(N-j)!\, j!} \tag{7.15}$$

[function **binomial(N,j)** *in Matlab, my program.]*

Definition 17 (Bernstein Basis Functions)

$$\mathfrak{B}_j(x;N) \equiv \left(\begin{array}{c} N \\ j \end{array} \right) x^j (1-x)^{N-j} \tag{7.16}$$

7.8 Bézier Curves

Notation: let $\mathbf{b}_i$ denote a set of control points in either two or three dimensions:

$$\mathbf{b}_i \equiv= \left(\begin{array}{c} x_i \\ y_i \end{array} \right) \qquad \text{or} \qquad \mathbf{b}_i \equiv= \left(\begin{array}{c} x_i \\ y_i \\ z_i \end{array} \right) \tag{7.17}$$

Then the explicit form of the Bezier curve is

$$\mathbf{b} = \sum_{j=0}^{N} \mathbf{b}_j \, \mathfrak{B}_j(t;N) \tag{7.18}$$

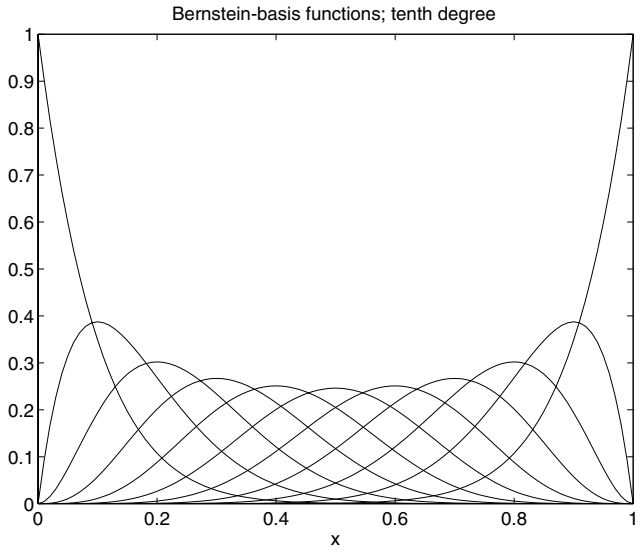


Figure 7.7: The Bernstein basis functions, $\mathfrak{B}_j, j = 0, \ldots 10$ of degree N.

where $t \in [0,1]$ is the continuous variable that parameterizes the curve. In words, each of the functions $(x[t], y[t], z[t])$ is approximated by a Bernstein approximation of the same degree.

The Bezier curve (and therefore the Bernstein polynomials) can also be evaluated recursively by the "de Casteljau" algorithm (Farin, 2001)

$$\mathbf{b}_j^r = (1-t)\mathbf{b}_j^{r-1} + t\mathbf{b}_{j+1}^{r-1}, r = 1, 2, \ldots N, j = 0, \ldots, N-r \tag{7.19}$$

Then

$$\mathbf{b} = \mathbf{b}_j^0 \tag{7.20}$$

Definition 18 (Control Points and Polygon) *The points $\mathbf{b}_j$ are called the "control points" of the Bézier curve; note that the curve interpolates only the two endpoints. The polygon formed by connecting the control points by straight lines in order is called the "control polygon".*

Theorem 3 (Bézier Curve Properties)

1. *Endpoint interpolation: the curve always passes exactly through $\mathbf{b}_0$ and $\mathbf{b}_N$.*
2. *Invariance under affine maps, which is any transformation that can be expressed as a matrix connecting (x, y, z) with (x', y', z') plus a translation.*
3. *Convex hull property: $\mathbf{b}$ lies in the convex hull of the control polynomial.*
4. *Linear precision: if the control points are evenly spaced on a straight line, then the Bézier curve is the linear interpolant between $\mathbf{b}_0$ and $\mathbf{b}_N$ and in fact interpolates all control points.*
5. *Variation diminution: no straight line (in two dimensions) or plane (in tree dimensions) intersects the curve more often than the control polygon. This property accounts for the shape preservation property of Bézier curves.*

Unproperty: Bézier curves are NOT invariant under projective maps although so-called "rational Bézier" curves can be constructed which do have this property.

The derivative of a Bézier curve is a Bézier curve:

$$\frac{d\mathbf{b}(t)}{dt} = N \sum_{j=0}^{N-1} \Delta\mathbf{b}_j \mathfrak{B}_j(x; N-1) \tag{7.21}$$

where Δ denotes the forward differences, i. e.,

$$\Delta\mathbf{b}_j = \mathbf{b}_{j+1} - \mathbf{b}_j \tag{7.22}$$

Drawing programs normally use cubic Bézier curves and create the curves by specifying the control points as explained in the next subsection. In industry, however, Bézier curves are also used to approximate curves that have been measured experimentally or from a clay model. To smooth out measuring errors, the usual strategy is a least-squares fit of a Bézier curve of modest degree, such as a quintic polynomial, to a much larger number of measured points (such as seventy points).

7.8.1 Corner Points and Smooth Points

Drawing programs such as Adobe Illustrator typically represent complicated shapes as *composite* Bézier curves of third degree. The overall shape is composed of a number of separate but connected pieces, each of which is a different cubic polynomial in a parameter t. Each piece consists of two endpoints, which are actually on the curve, plus two additional control points which are near but not on the cubic curve.

If the right endpoint of one segment is the *same* as the left endpoint of the adjoining segment, then the composite Bézier curve will automatically be continuous. The slopes may or may not be continuous. At a "corner point", in Illustrator parlance, the slopes of two adjoining segments are different. At a "smooth point", however, the slopes must match so that the composite curve is continuous.

It is proved in many CAGD books such as Mortenson(1999, pg. 266) that the slope of a cubic Bézier curve at an endpoint is equal to that of the straight line connecting that endpoint with the nearest control point. This implies that one can create a "smooth point" at the endpoint which is common to cubic Bézier curves by constraining the two control points nearest the endpoint to be collinear, that is, to lie on the same straight line.

In Illustrator, when one clicks on an endpoint where two neighboring curves join, one sees two little dots — the nearest interior control points — connected by straight lines to the endpoint. One can edit the shape of the curve by dragging on either of the control points; the curve will change shape even though the endpoint is fixed (unless it is the endpoint, rather than the interior control points connected to it, that is dragged.) When the joining point as been flagged as a "smooth point", the two interior control points are forced to lie on the same straight line so that when one is dragged, the other control point rotates so as to remain on the line formed by the other, dragged control point and the smooth point (Fig. **??**). At a "corner point", the two adjacent interior control points are completely independent. When the left interior control point is dragged, the shape of the left segment changes, but the curve to the right of the corner point is completely unaltered. In contrast, when a curve in the vicinity of a "smooth point" is edited, either by dragging the point itself or the two neighboring control points, then *both* segments which meet at that smooth point are altered simultaneously.

In any even, no more than two segments of a composite Bézier are moved by dragging a control point. One could represent a complicated curve by a Bézier shape of high polynomial degree, which might seem simpler than using a composite of cubic shapes. However, it is easy with composite cubics to change a point on the curve from "smooth" to "corner" or vice versa merely by removing or adding the constraint that the other interior point must stay collinear with the one which is dragged. Furthermore, when a composite curve is edited, only one or two parts of it are altered. If a single high degree curve were edited, moving one control point would change the shape everywhere. It would require vast amounts of fiddling to get the shape just right because parts that were already acceptable would be distorted by adjustments made to fix the shape elsewhere.

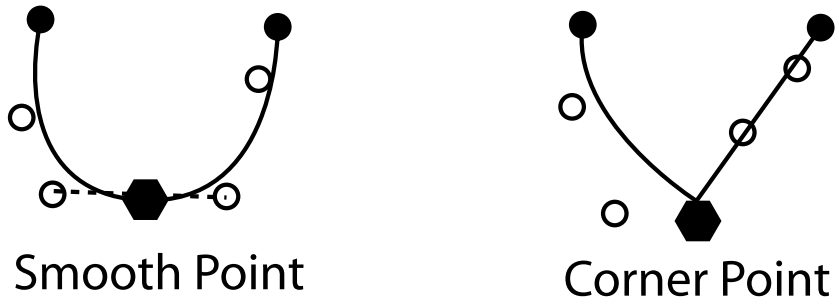


Figure 7.8: Composite Bézier curves in which two cubics are joined by a smooth point (left) or corner point (right). The control points which are common to two segments are marked by hexagons; the other endpoints are filled circles. The interior control points are hollow circles. For the smooth point, a dashed line connects the two control points that are constrained to lie on the same line so that the slope is continuous where the segments join.

7.9 Texture Mapping

Every colored surface is a pattern of patches. Each patch must have its own color. If the input to a surface-making subroutine contains only three matrix arguments **X, Y, Z** specifying the Cartesian coordinates of the points of the surface, the routine must create a fourth matrix **CData** to store the color data associated with each patch. In a function like Matlab **surf**, the color values are initially slaved to the values of Z so that the largest positive values of z are coded dark red, and the points of most negative z are dark blue. However, the colordata array can be overwritten to associate an arbitrary color image with the surface. Such overwriting of the colordata of a surface is called "texture-mapping".

Although we shall illustrate the method using Matlab, texture-mapping is a very easy trick — simply overwriting a matrix that the surface-maker had to generate anyway — that is widely employed in computer graphics. The listing below describes a Matlab example. The "texture" is the colordata of a pseudocolor plot (Fig. 7.6) which is generated in the first five lines. The sixth line, **cmap=get(surfacehandle,'CData');**, extracts the colordata matrix that was generated by the pseudocolor plot and stuffs into a matrix called **cmap**. The next three lines plot the surface of sphere in three-dimensional space using Matlab's built-in sphere command. The final line is the other essential component of texture-mapping: **set(spherehandle,'FaceColor','texturemap','CData',cmap);** modifies the colordata (**'CData'**) of the sphere plot by replacing it with the colordata of the pseudocolor image.

The **'FaceColor','texturemap'** informs the plot that its colordata matrix has been altered. It also triggers a useful feature of Matlab texture-mapping: if the new **CData** matrix is different in size from that originally generated by the surface plot, Matlab automatically performs the necessary interpolation. This is in fact true of our example where the plot of the sphere used only 625 patches whereas the pseudocolor plot had more than 20,000.

Unfortunately, most Matlab routines do not generate surface objects. It is possible to texture-map line graphs, but these must be saved as Encapsulated Postscript (EPS) files, then converted to TIFF or JPEG formats using Adobe Illustrator or Graphic Converter, and loaded back into Matlab through the **imread** command. Complicated surfaces like a sphere are most readily perceived if a surface mesh is included, but this makes it hard to identify the data curves. If the grid lines are converted to grey dotted lines, however, then line graphs on a sphere or other complicated surface can be effective.

```
hlong=2*pi/200;   hlat=pi/100;   % Part 1: PCOLOR
[longitude,latitude]=meshgrid(-pi:hlong:pi, ...
                              -(pi/2):hlat:(pi/2));
dipole=latitude .* sech( 4*longitude.^2 + 4* latitude.^2);
surfacehandle=pcolor(longitude,latitude,dipole);
cmap=get(surfacehandle,'CData'); % Part 2: Get ColorData
\[Xsphere,Ysphere,Zsphere\]=sphere(25);  % Part 3: SPHERE
spherehandle=surf(Xsphere,Ysphere,Zsphere);
axis equal;    axis off;
        % Part 4: TEXTUREMAP
```

```
set(spherehandle,'FaceColor','texturemap','CData',cmap);
```

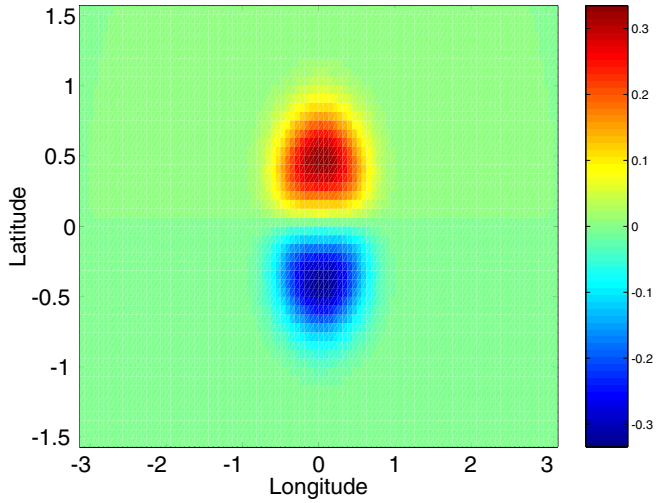


Figure 7.9: A pseudocolor plot in the usual flat, two-dimensional representation as a function of latitude and longitude. The next figure shows the same image texture-mapped to the surface of a sphere.

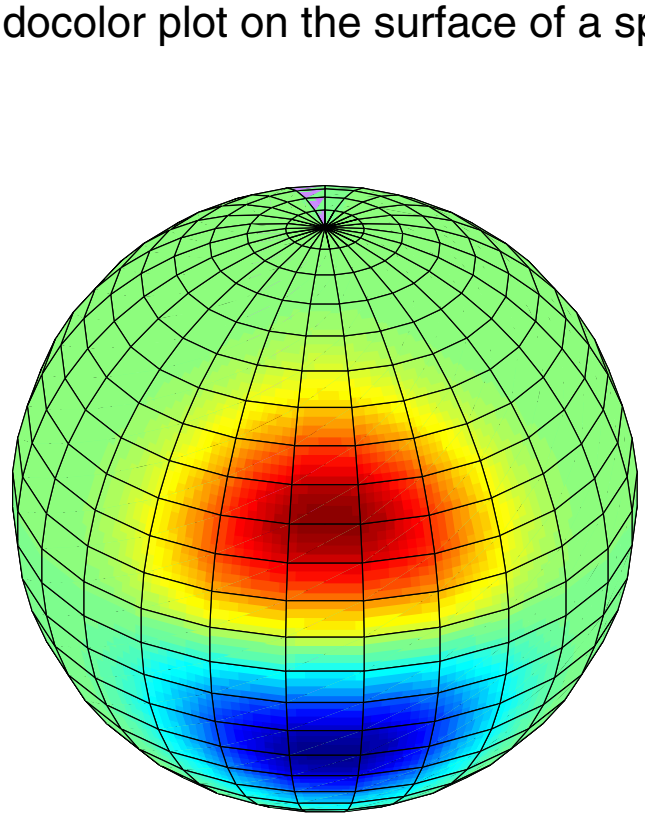


Figure 7.10: A sample of a pseudocolor plot of a function of latitude and longitude which was texture-mapped onto the the surface of a sphere.

7.10 Line Graphs on Surfaces

It is possible to graph ordinary functions of one coordinate versus another on a surface by texture-mapping, but this is awkward and inefficient. In Matlab, the line graph must be saved as an EPS file, converted to a TIFF or JPEG file, read in as a Matlab **image**, and then texture-mapped. The syntax is different in other graphics libraries, but it is always inefficient to convert a line graph, which is simply a pair of vectors, into an image, which is a two-dimensional array.

A better procedure is plot the line graph directly on the surface. In order to plot the surface, one must have a formula that converts a two-dimensional array (representing a grid of points on the surface) into three two-dimensional matrices which give the Cartesian coordinates of each of these points on the surface. For example, use latitude and longitude to parameterize the surface of a sphere:

$$\theta_i = -\frac{\pi}{2}\left\{1 + 2\frac{i-1}{M-1}\right\}, \qquad i = 1,\ldots,M, \qquad \lambda_j = -\pi\left\{1 + 2\frac{j-1}{N}\right\}, \qquad j = 1,\ldots,N, \tag{7.23}$$

The surface of the sphere is defined by connecting the points

$$x_{ij} = \rho\,\cos(\theta_i)\cos(\lambda_j) \tag{7.24}$$
$$y_{ij} = \rho\,\cos(\theta_i)\sin(\lambda_j) \tag{7.25}$$
$$z_{ij} = \rho\,\sin(\theta_i) \tag{7.26}$$

where ρ is the diameter of the sphere.

The three-dimensional curve of a function $\theta(\lambda)$ is obtained by applying the same transformation formulas which define the sphere itself:

$$xx_j = \rho\,\cos(\theta(\lambda_j))\cos(\lambda_j) \tag{7.27}$$
$$yy_j = \rho\,\cos(\theta(\lambda_j))\sin(\lambda_j) \tag{7.28}$$
$$zz_j = \rho\,\sin(\theta(\lambda_j)), \qquad j = 1,2,\ldots,P \tag{7.29}$$

In Matlab, plot the sphere with **surf(x,y,z)**, keep this with **hold on**, and add **linehandle=plot3(xx, yy, zz)** to plot the data curve. Fig. 7.8 is a sample.[1]

To plot any other sort of surface, we must have an algorithm . That same transformation can then generate the points of a data curve in three-dimensional space which will hug the surface. Texture-mapping is unnecessary for curves; we need only to map surface-to-surface.

[1] In Matlab, it is helpful to use a radius for the data curve which is slightly larger than for the sphere so the surface does not obscure parts of the line graph.

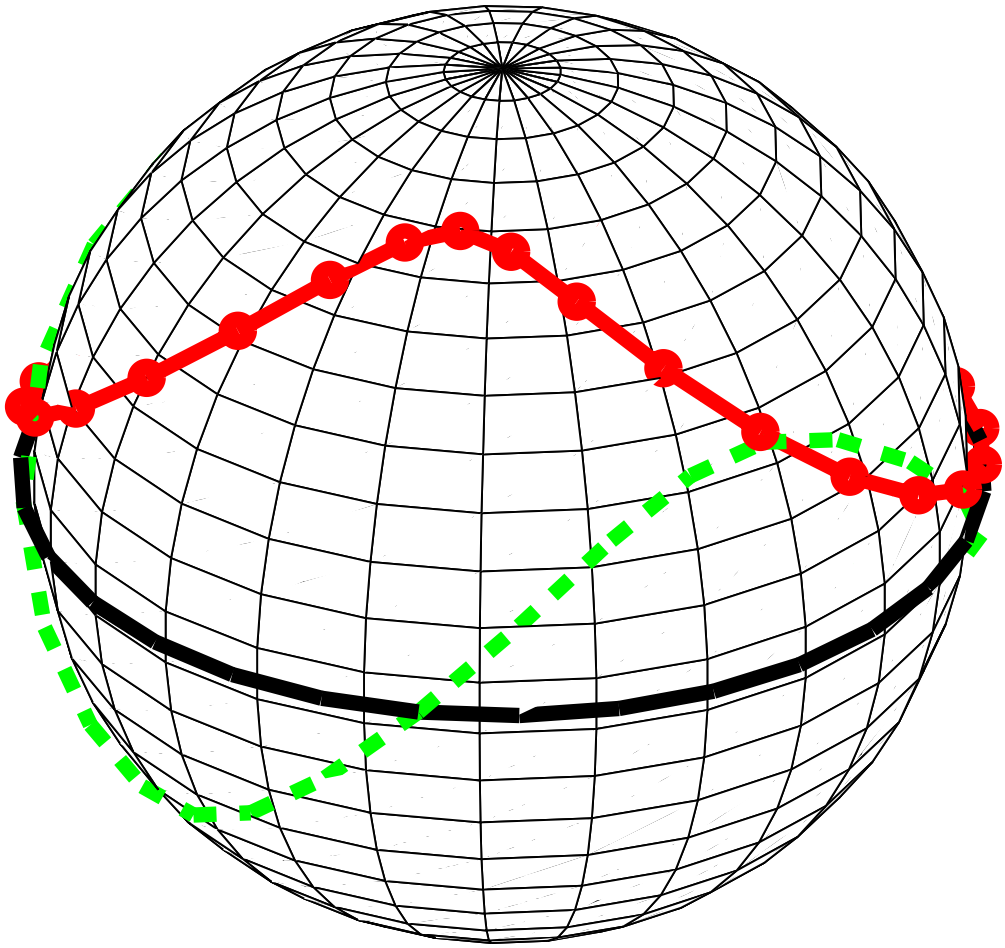


Figure 7.11: Functions of latitude as a function of longitude are shown on the surface of a sphere as a red solid line with disks and a cyan dashed line. The equator is the heavy solid black line.

7.11 Drawing versus Painting or Bitmap versus Vector

There are two fundamental approaches to graphics: bitmap and vector. Bitmapped graphics represent an image as a two-dimensional array: one element for each screen pixel. The disadvantage of bitmap graphics is that it requires a large file: with a 1000 x 1000 pixel screen, one must specify a million numbers just for a single black-and-white image. Color requires 8 bits to 24 bits per pixel. Individually picking each pixel would take a lifetime; bitmap graphics are natural for photographs and complicated textures that can be generated by algorithm.

Bitmapped graphics are also natural for PAINTING by computer. In a paint program, the movement of the mouse or graphic pen represents an imaginary paintbrush of a certain width. As the mouse moves, all pixels within the width of the brush are altered to the color of the brush. Thus, a single stroke may alter the colors of thousands or ten of thousands of pixels. The program allows the user to change the width and color of the brush and often to add special effects such as leaving uncolored spots within the paint-swath so as to lay down a texture rather than a solid band of color.

Vector graphics represent an image as a set of instructions, such as "draw a line from (1,0) to (0,1) of 0.5 points thickness in the color red". Vector graphics has two advantages over bitmaps. First, when the figure consists of only a few line segments, the instructions to draw them can be stored in a miniscule fraction of the number of bytes needed for a bitmap. Second, vector graphics are hardware-independent. A diagonal line of 2-point width in green is an unambiguous instruction whether the output device is a screen with 72 dpi resolution or a printer with 300 dpi resolution. The disadvantage of vector graphics is that as the image becomes more complicated, both storage and display/print time will increase until both exceed the cost of a bitmap: a million on/off commands in a bitmap array are much cheaper than a million vector instructions, each of which describes beginning point, endpoint, type of element, color, width and so on.

Both types of graphic representation have flourished for many years because neither is superior to the other for all situations. The dichotomy was reflected in the early days of the Macintosh by the existence of two competing illustration programs: MacPaint, which was all bitmaps and brushes, and MacDraw, which was all vector graphics, lines and polygons.

In recent years, as programs have become more sophisticated, these two poles of bitmap/vector, MacPaint/MacDraw, have cautiously drawn closer, rather like the warring parties in Northern Ireland. However, it is very hard, technically, to combine such disparate elements. Most programs are firmly in one camp or the other, and show only halting and limited support for the other metaphor.

Matlab and most scientific software is vector-based. A linegraph is not a two-dimensional array of pixels but rather a one-dimensional vector of points connected by line segments.

Matlab does support an **image** datatype which is a sort of resolution-independent bitmap. An image might be imported as a 200 x 200 array. On a million pixel screen, Matlab will display the image by using the same element for a square of twenty-five pixels, assigning a color to the pixel by using the number in the image array as an index into whatever colormap is in effect. However, images are not terribly well integrated into the rest of Matlab. For example, if the image is in the low-storage form, a one-byte integer for each element, then no arithmetic operations on array elements is allowed. If the image is stored in Matlab's default double precision form, one may easily run out of memory unless the image arrays are small and few. Texture-mapping is a "paint" operation, but compared to a real painting program, Matlab is a dog of few tricks.

Adobe Illustrator is a vector-graphics program. It also supports textured fills of

polygons and even a paintbrush tool, but the action of the paintbrush is converted to a curve enclosing the swath of a brush. Adobe Photoshop, in contrast, is focused on bitmap graphics. The vector/bitmap duality has put Adobe in competition with itself!

KidPix Studio is a paint program, much beloved of children. Its pictures are always drawn at the 72 dpi screen resolution of Macintosh computers. However, it does have tools for drawing lines, circles, and polygons just like a vector-based drawing program.

Scientific graphics is mostly vector graphics; the exception is image-processing, such as the output of a CAT scanner or an astronomical camera. In the rest of this chapter, we shall concentrate primarily on vector graphics. However, we must keep the draw/paint dichotomy in mind because these two functions can never be entirely separated.

7.12 Drawing Programs: Tools

7.12.1 Selecting Objects

Fig. 7.10 illustrates the basic toolbox palette for a representative drawing program, Adobe Illustrator, which is available for both Macintosh and Windows PC. Illustrator is primarily a vector-graphic drawing program; the paint/draw-bitmap/vector duality is reflected in Adobe's other major program, Photoshop, which is primarily for bitmap images. Illustrator has a few paint-like tools, however, which reflects the partial convergence of painting and drawing.

In a drawing program, a figure is a collection of elements: line segments, continuous curves, polygons, rectangles and text boxes. To move or modify an element, one must first SELECT it. The top two tools on the left palette are both selection tools (arrows).

7.12.2 Outline View and Preview View

Illustrator allows two different ways or "views" for displaying an illustration on the screen. In "preview" mode, what-you-see-is-what-you-get. Unfortunately, in preview mode, it can be very difficult to properly select and edit objects when the screen is full of many ellipses, rectangles and other elements. In outline view, objects are seen as skeletons, and the selection tools work a little differently than in preview mode to make it easier to select just the object you wish to edit.

The solid black arrow is the "selection tool"; it might also be dubbed the "select-to-move-the-whole-thing". When the black arrow has been clicked, and the cursor is moved by the mouse, a little black square will appear next to the cursor arrow when the cursor is sufficiently close to an object to select it. The definition of "sufficiently close" is different between "preview mode" and "outline mode". In preview mode, if you move the cursor onto either the side or the interior of an ellipse (or whatever), the black square will appear. In outline view, most of the interior of the object is neutral, and one can select an object only by moving the cursor very close to (i) one of its sides or (ii) the little "x" that marks the centroid of the figure in outline view only. (The centroid "x" is invisible in preview mode except when the object has already been selected, and the object is not filled with a color.)

In either mode, clicking the mouse when the black square is visible will select the object. In either preview or outline mode, selecting an object means moving the cursor close to it until the black square appears next to the cursor arrow and then When the rectangle (or whatever) is selected, a rectangle circumscribing the object will appear in blue, and there will be blue squares at the corners and the midpoints of this rectangle.

If the cursor is close to the centroid of a selected object — the centroid is marked by a tiny "x" in the middle of the object like the pirate gold on the map in *Treasure Island* — then the arrow changes to a guitar pick. If the mouse is then dragged, the whole object moves. Thus, to move an entire object:

1. Select the object (black arrow)
2. Move cursor arror to the "x" until the cursor becomes a wedge or guitar pick.
3. Drag the mouse

If one of the hollow rectangles is dragged, then only that side, or that diagonal (if one of the corner points) will move. Thus, the arrow tool can both move a shape, and also deform that shape. One has to be careful to move objects by dragging the little "x" in the middle only.

When a rectangle is selected by clicking the selection tool and then clicking when the mouse cursor is over the rectangle, then dragging with the mouse will move the figure to a new location.

One must be careful, however. When a rectangle is selected, for example, one will see hollow boxes in the middle of each side. When the cursor is over one of these, it will changes to a two-sided arrow. One can then drag to move that side of the figure.

It is difficult to move the mouse with the ultrahigh precision that is sometimes needed for drawing. Illustration and its cousins allow the artist to constrain the mouse in a couple of different ways when needed. First, holding down the shift key will force the figure to move in a purely vertical or horizontal direction, whichever is closest to the actual direction of the mouse motion. (Some programs allow the shift key to constrain to moving along that angle which is the multiple of 45 degrees that is closest to the mouse motion). Second, one can use a pull-down menu to temporarily impose one-pixel-at-a-time motion for very fine control.

Most graphic objects also consist of multiple "anchor" points; for a rectangle, these are just the four corners. One often needs to move one of these anchor points to change the shape of a curve or figure. The "direct selection tool" (hollow arrow) might be called the "anchor point selection tool". By clicking this tool and then near one corner of the rectangle, one can move just the selected anchor point, distorting the rectangle into a general quadrilateral.

However, one can also click with the white arrow in the middle of a line segment. One can then drag with the white arrow to move the segment, both endpoints moving in parallel. So the "white arrow" is the "anchor or endpoint selection tool".

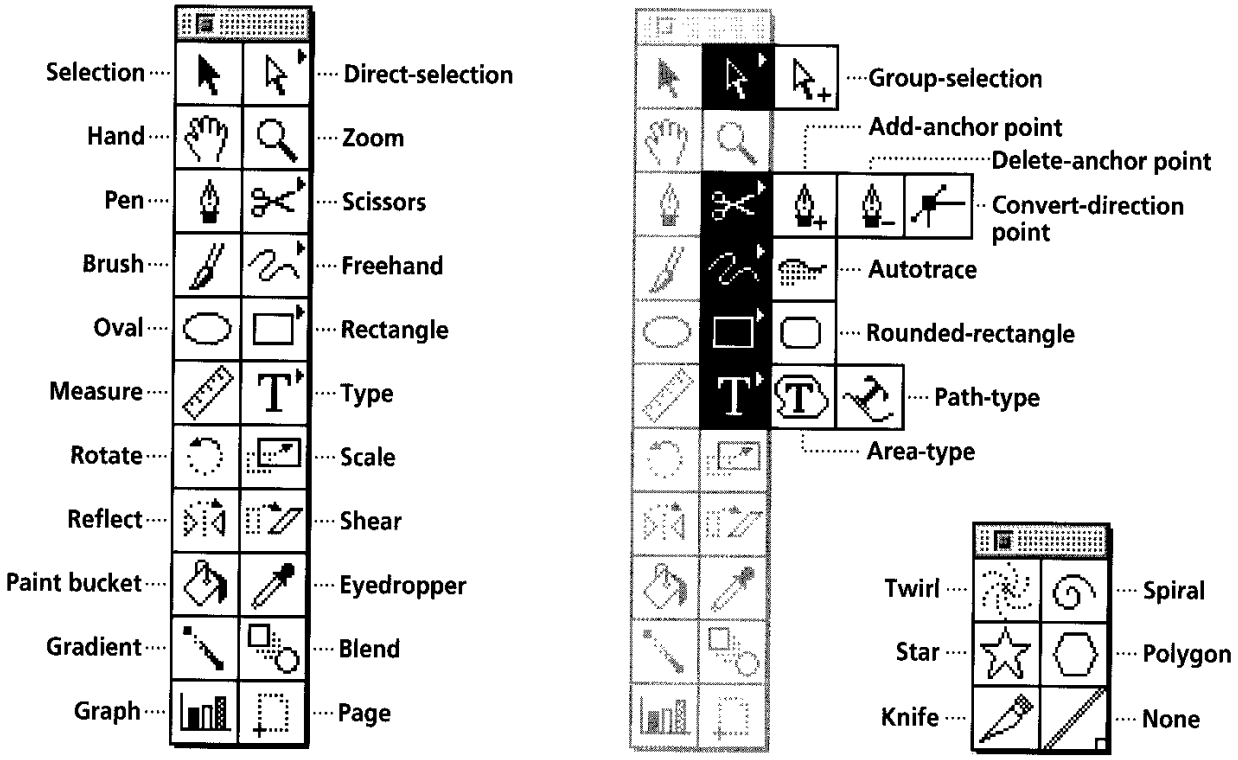


Figure 7.12: Basic tool palette in Adobe Illustrator, version 6. The tools at the left are always visible; one activates a tool with a mouse-click. The "hidden" tools become visible when the mouse is held down and dragged, beginning over a default tool. The special tool palette on the right may be either visible or invisible at the user's option, but the tools themselves are always available. This palette and a wide variety of other options, setting and tools can be brought to the screen and applied by activating drop-down menus from the menu bar at the top of the screen.

The “direct selection tool” has its rather bland and general name because it can be used to do more than select a single anchor point. If one clicks on a line segment connecting a pair of anchor points, then the entire segment is selected. This makes it possible to “move a wall” by dragging such as to stretch a square into rectangle without the need to individually move a pair of corners.

The hollow arrow with a small cross, available by dragging the direct-selection tool, is the “group selection tool”. The reason that this is needed is that illustration programs allow one to freely group (and ungroup) graphical elements.

For example, suppose one has drawn a flower using separate curves for the stem, petals, stamen and pistil and markings on the petals. If one needs to rearrange the figure by moving the flower, one could of course move each element individually, but this would be torture. After each part had been moved to the general location, one must tweak each element into proper location with respect to the others, perhaps only to realize that the flower is not quite aligned with other parts of the figure, and the whole assembly needs to be moved again.

A far better way is to join the elements as a group, which can then be moved as a single entity. The procedure is simple: hold down the shift key and click on the elements of the group, one by one, until each is highlighted. Then, pressing the Apple key and “g” simultaneously will unite the elements in a group. The group can then be moved with a single click-and-drag. After the move, the elements can be ungrouped with a single keyboard command so that individual elements can be edited.

Yet another way to select graphical elements is by using a “selection marquee”. If one drags after selecting one of the selecting tools, a dotted box will appear. When the box is released, all the elements in the box are selected (Fig. 7.11.)

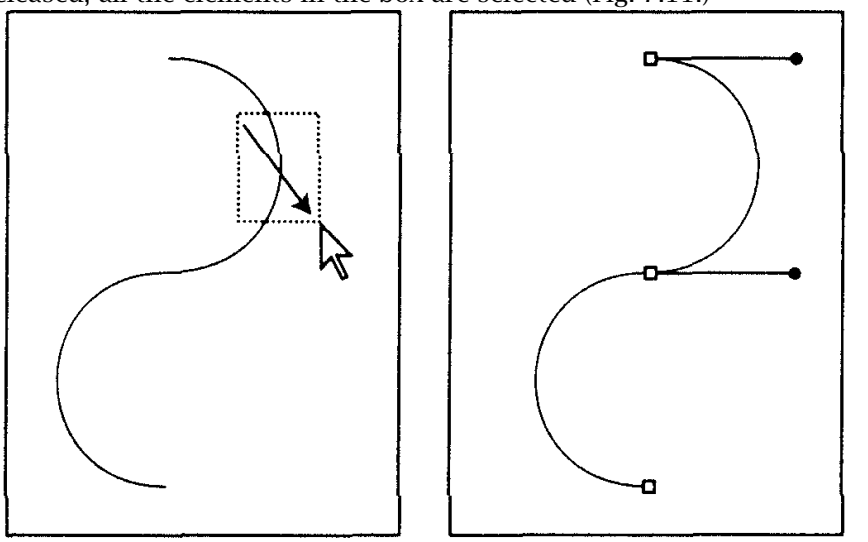


Figure 7.13: Using a selection marquee: the dotted grey box is drawn by dragging with an arrow-shaped cursor after clicking one of the selection tools.

7.12.3 Views and Layers: Selecting in a Crowd

Selection of an object sounds simple, but there are some complications. One is that when the drawing has many elements, they may overlap so much that it becomes very hard to select an individual line segment or polygon. Drawing programs provide a couple of strategies that help. First, Illustrator offers a "preview" mode which shows the elements in a skeletonized-form with all color fills and textures fills omitted. It is much easier to select objects in preview mode than in the "artwork" view, which shows the objects as they will appear when printed.

Second, Illustrator and other programs allows the figure to be organized into LAYERS. If overlapping elements are created in different layers, then one can select the element easily after first choosing the proper layer: the selection tools ignore all elements that lie in different layers. It is trivial to toggle between artwork and previews views and between one layer and another because the program was designed thus: switching views and layers is very valuable to the artist, even though it has nothing directly to do with how any of the elements actually look.

Layers can be named. In the layers menu, one can click on the layer titled "hexagon" to easily select and edit a six-sided figure. If the hexagon is on a layer by itself, one does not have to worry about accidentally selecting other objects, or inadvertently modifying other objects when the solid border of the hexagon is changed to dashed lines, for example.

7.13 Layers

In the early days of animation, each frame was drawn from scratch by hand. Since one minute of animation requires 1440 frames at 24 frames/second, all-in-one animation was incredibly slow and expensive. By the mid-1920's, a better way had been developed. Elaborate backgrounds were painted onto paper, but only once per scene. The characters and moving objects were painted onto sheets of transparent celluloid. Animation costs were reduced by an order of magnitude because only a small fraction of each scene was redrawn for each frame. If multiple objects were moving in a given scene, each was drawn on a separate transparent sheet. If a character was motionless, watching a bouncing ball, then it might not be necessary to do any new drawing to create several seconds of animation. The "cel" with the ball would simply be shifted slightly by the camerman between each click of his camera, filming the composite of background-plus-multiple-cels on frame at a time.

In this process of layering was extended still further. In the *Space Angel* cartoons of the 1960s, the mouths were drawn on cels separate from the bodies. A character could speak by merely moving his mouth while the body cel remained unchanged. Once a small library of cels showing the mouth in different positions had been drawn, the cameraman could film a dialogue scene without requiring any additional drawings.

Layering isn't heavily used in other forms of non-computer media, although it is common for artists to lay down a layer of tape to protect parts of the drawing before airbrushing the background, for example. However, computer illustration and image-processing programs such as Adobe Illustrator and Adobe Photoshop use layers as heavily as animated cartoons.

Why layer? Some answers follow:

1. Easy selection and editing of objects in a crowded drawing.
2. Toggling layers "visible" and "invisible" allows one to temporarily blank out groups of objects so that the others can be seen more easily.

3. Scan-and-trace: put the scanned image and separate layer and then delete the layer when the trace is down.

4. Isometric drawing and drawing with grids or guides: put the grid on one layer and the rest of the drawing on other layers, and then delete the grid/guide layer at the end.

A pop-up layer submenu allows one to do the following:

1. Assign a name to each layer (the defaults are “one”, “two”, etc.)

2. Mark the layer as modifiable or frozen

3. Mark the layer as visible or invisible

Each layer has a different selection color; this helps to avoid accidentally modifying an object on layer three when one really intended to edit an element on layer two. Freezing all layers except the one you want to work on is a big help, too.

The layers palette menu, which is accessed by dragging the little triangle in the upper right corner of the layers menu, allows one to create, destroy or reorder layers.

Layers are valuable when TRACING imported images. The import can be stored on layer one; the trace can be performed on layer two. The superposition of two (hopefully similar) images can make it hard to check the trace but one can toggle the traced image layer visible/invisible through the layers menu to see the trace in isolation. When the trace has been completed, the layer with the imported image can be simply erased. (Parenthetically, note that the trace, composed of vectors, is usually a small file compared to the imported bitmap image.)

Layers are equally valuable when drawing on a grid. Illustrator allows one to create non-printing “guides”. These not be deleted.

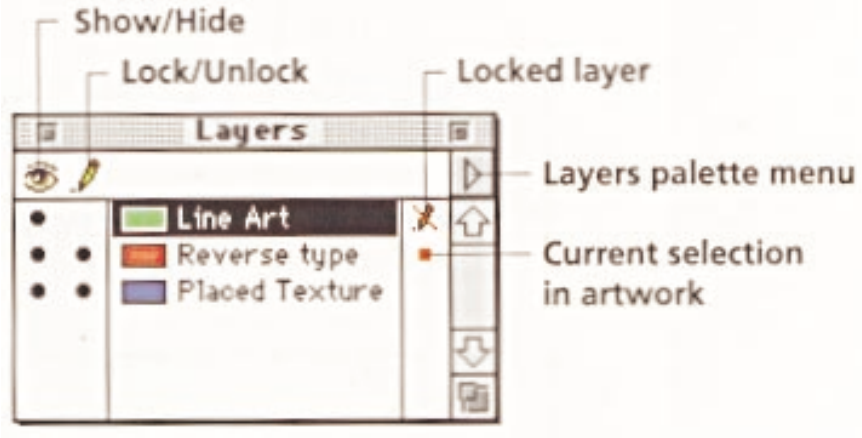


Figure 7.14: Illustrator Layer Menu. The tickmark in the left column specifies whether the layer is *visible*. (There is a stylized icon of an eye above this column. The marks in the second column from the left (under the icon of a pencil) denote whether the layer is modifiable or not. To unlock a frozen layer, one must go to the layer menu and click in this column so that a mark appears next to the layer in this column, and then one can modify objects in this layer. The wide central column gives the names of each layer. It is often convenient to override the default names, which are merely numbers, to give each layer a label which represents its function. The colored rectangles next to each layer name gives the selection color for that layer, which by default is a different color for each layer.

Another use of layering is to avoid inadvertent changes: when many elements are crowded in a small place, it can be very hard to choose the desired line segment and successfully avoid accidental alteration of others. When the elements are in different layers, however, one can work only on layer at a time. Furthermore, other layers can be made invisible, which may make it much easier to see what one is doing.

7.14 Zoom and Grab

The next two tools on the Illustrator palette draw nothing, but merely make the image easier to see. The zoom tool is invariably depicted as a magnifying glass. Click the tool and then click the image, and the screen will be filled with a magnified image which is centered on the location of the mouse-click. Shift-zoom is the anti-zoom, reducing the image so that one can see the whole page again. Most drawing programs support extremely high levels of magnification so that one can literally manipulate individual pixels at 300 dpi resolution if necessary.

One drawback of zooming is that sometimes the desired elements will disappear off the screen at high magnifications. The "grab-hang" allows one to re-center the image; click and drag on the screen, and then the image will appear to shift position in the direction of the mouse motion.

7.14.1 Pens and Brushes

Most Illustration programs have at least three tools for drawing lines and curves. KidPix Studio, for example, has a line tool for drawing line segments plus a brush tool for drawing curves. When either tool is activated, a pop-up menu appears at the bottom of the screen, offering a choice of line widths and color intensities: dark, medium, light or an almost transparent wash. The KidPix brush also offers a choice between a square brush (which leaves sharp edges at the ends of each stroke) or a rounded brush.

Adobe Illustrator offers a similar range of tools, but with more sophistication and a lot more options. The pen tool is the means to draw straight lines. If one clicks on the pen tool, and then clicks on N points on the screen, the points will be connected by straight line segments.

The freehand drawing tool, just below and to the right of the pen, is for drawing curves. As the mouse is dragged across the screen, Illustrator draws a smooth continuous curve with continuous slope that mimics the motion. (Fig. 7.13.) Behind the scenes, Illustrator is actually fitting the mouse or pen motion by a Bezier curve.

Tracking the motion to the nearest pixel has disadvantages (usually): the Bezier curve needs lots and lots of control points, which gives a curve that is hard to edit and takes up a lot of storage and display time, and the zigs and zags may reflect small-scale unsteadiness of the human hand more than the bold strokes intended by the artist. Illustrator therefore allows the artist to specify how closely the freehand curve tracks the mouse motion. The default tolerance is keep the curve within two pixels of the mouse motion. Fig. 7.14 shows that reducing the tolerance reduces the number of control points.

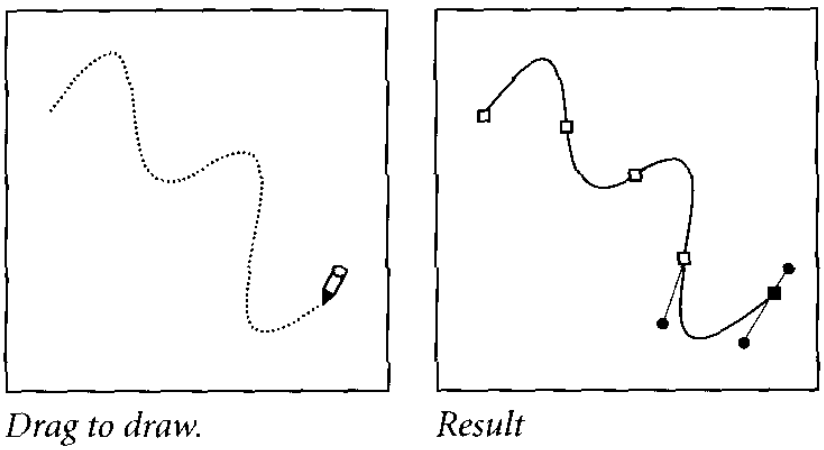


Figure 7.15: Schematic of the Illustrator freehand drawing tool. Left: the motion of the mouse. Right: the curve laid down in response to the mouse movement; the white disks are its ancho points. The direct handles of the last two anchor points, which end in solid disks, are also shown.

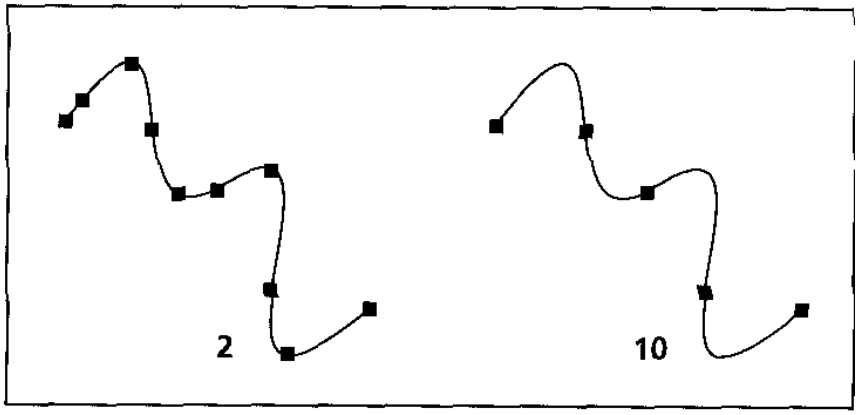


Figure 7.16: Two curves drawn by the freehand drawing tool in Illustrator with the same mouse motion, but different freehand tracking tolerances. Left: track-within-two-pixels (default). Right: track-within-ten-pixels, which halves the number of control points.

Rather confusingly, the pen tool can also draw curves. If an anchor point is dragged after being created, then the dragging will make the point into a "smooth point" with direction lines parallel to the direction of the dragging. (Fig. 7.15). One need not drag all the way to the next anchor point; indeed, the best practice is to drag only a short distance and then release the mutton button and move to the next anchor point, which is created by clicking.

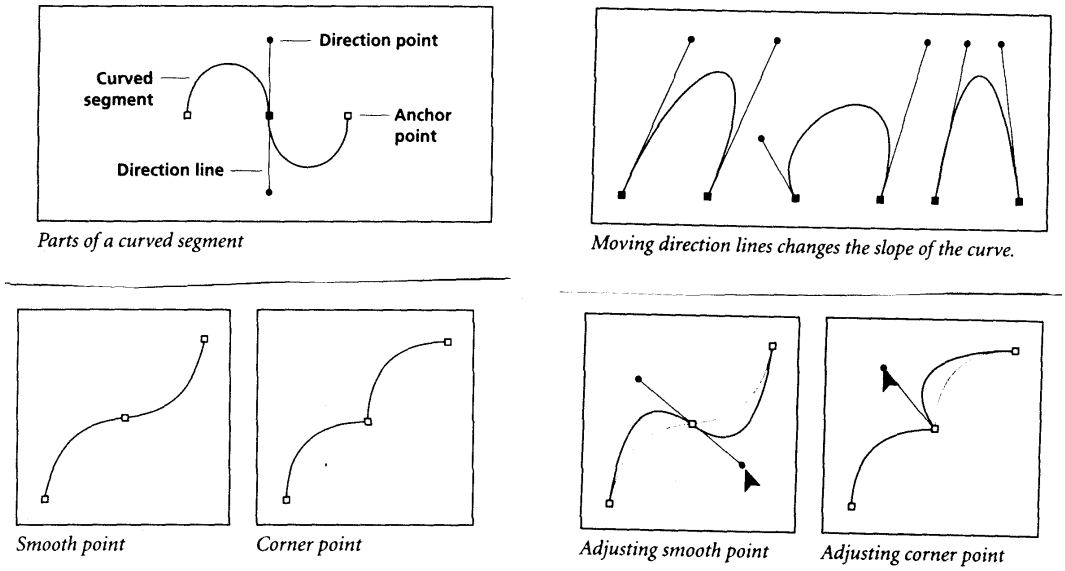
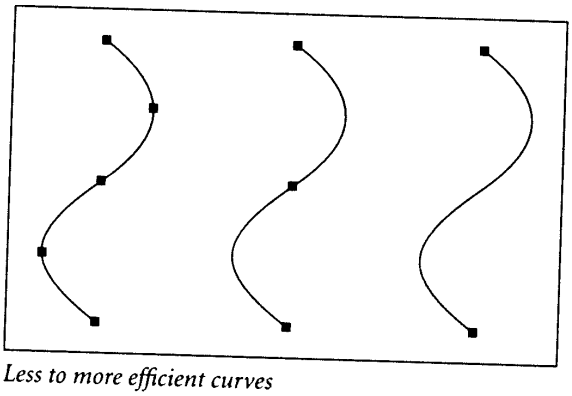


Figure 7.17: Upper left: schematic of anchor points and direction lines. Upper right: three non-identical curves through the same pair of points; the shape varies because of variations in the direction lines through the anchor points. Lower left: a visual definition of the difference between two species of anchor points: "corner" points (where the slope of the curve is discontinuous) and "smooth" points. Lower right: adjusting smooth points and corner points; a corner point has only one direction line whereas a smooth point has two direction lines that can be independently varied.

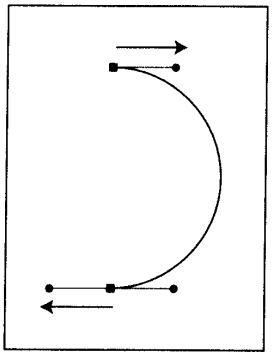
It is terribly easy for the novice to click-and-click without dragging in between, and thereby making the first anchor point a "corner point" where the slope need not be continuous and the two anchor points are connected by a straight line. The reason for this seemingly dumbhead decision is that many real shapes are a mixture of straight lines and curves. The Illustrator pen tool can mix both angles and smooth curves in a single connected element. An experienced user may be driven to drink before mastering the pen, however.

Fig. 7.16 schematically illustrates the finer points of Illustrator penmanship. It is most efficient to use a SMALL number of anchor points whenever possible unless one really, really needs to follow an intricate shape. Anchor points are like babies: are you make 'em, they turn out to need a lot of love. Adjusting the anchor points of a curve with many so that the direction lines and locations give a curve that has no obvious kinks can be as hard as teaching a centipede to tap dance.

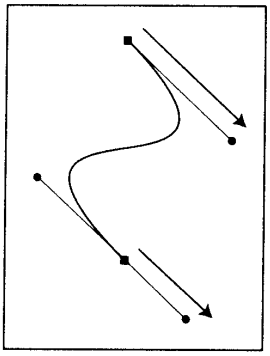
To create a smooth, half-sine-wave loop between two anchor points, one must drag first one way and then the other, which is rather counter-intuitive. Dragging in the same direction twice gives an S-shaped curve. This sort of between-the-anchor-points kink is usually an accident, but it does demonstrate the power of the direction lines to quite drastically alter the shape of a curve.



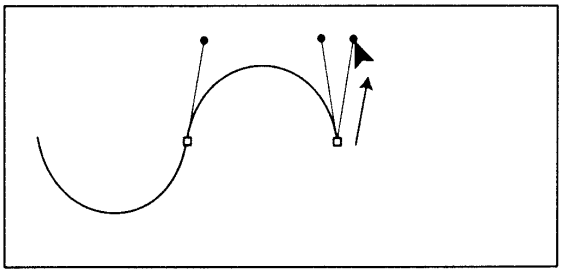
Less to more efficient curves



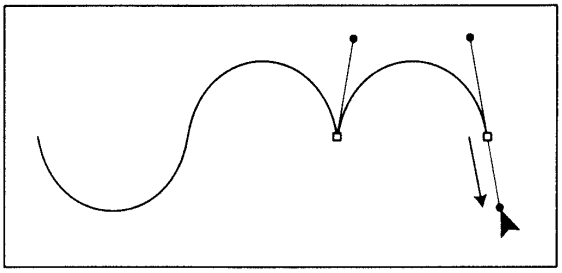
Drag in opposite direction to create smooth curve.



Drag in same direction to create "S" curve.



Press Option key and drag in direction of curve . . .



and then release Option key and drag in opposite direction.

Figure 7.18: Upper left: in using the pen (or freehand) tool, "less is more". Upper right: to draw a smooth, parabola-like curve between two anchor points, drag in opposite directions at the two points. Dragging in the same direction produces an S-shaped curve. Bottom pair of graphs: inserting a corner point into an otherwise smooth curve.

To add to the confusion, Illustrator and KidPix have a paintbrush. The color and width of the brush can be varied to lay down broad strokes of color. The "caps" (i. e., the ends of the brush-curve) can be varied to be round or square. In Illustrator, the paintbrush can be slaved to a pressure-sensitive pen-and-graphics-tablet so that the width of the brush is proportional to the pressure, just as in a paintbrush. One can also activate a "calligraphic" feature so that the width varies smoothly with the angle of the curve, in imitation of calligraphic writing. [2] Fig. 7.17 is a sample of the brush's capabilities.

The boundaries of a brushstroke is a Bezier curve, dotted with a few anchor points. These can be manipulated, just as with the output of the pen tool or the freehand drawing tool, to alter the shape of the brushstroke.

7.15 Stroke and Fill

In all forms of art, some objects are shown only in outline whereas others are filled with color. Most drawing programs therefore have tools for both drawing outlines and filling polygons and other outlined shapes. In Illustrator, almost every tool has the ability to apply both a "stroke" and a "fill". The "stroke" is the outline of the shape drawn; the "fill" is what goes inside the stroke. The pen, freehand drawing tool, brush and special tools for polygons, stars and spirals all support both stroke and fill.

For each tool, one may omit either the stroke or the fill. The colors of stroke and fill can be different. Illustrator allows both the stroke and fill to use patterns: patterns of confetti, gradients of smoothly varying color and more complicated shapes and textures such as a brick wall, transparent waves and so on. One can create new textures and patterns in Illustrator which can then be applied in exactly the same way as the generic textures supplied with the program.

The metaphor of the brush tool as a computer realization of a paintbrush breaks down when the brush tool is used to apply a swath of texture in a single mouse movement. An oil painter would have to individually paint each line segment for a pattern of confetti, but the computer artist does not. An oil painter would have to paint the fill in red, then change brushes to outline it with a stroke in black. Again, after the paint palette has been used to specify the stroke and fill colors as black and red, the brush tool can apply both the filling and the outline in a single movement.

The paint palette offers the option to change colors, line widths, fill patterns, and almost every other aspect of a curve or patch created with the pen, freehand drawing tool, paintbrush, or one of the specialized polygon, star and spiral tools. Fig. 7.18 shows the choices available. The palette can be hidden or displayed explicitly at the user's option; when invisible, it can always be accessed as a pull-down menu item from the menu bar at the top of the screen.

The little colored squares in the upper left of the palette provide access to a wide range of pre-defined colors. To make the stroke of a freehand curve green, for example, one first clicks the box labelled "stroke" in the upper left, just above the color swatches, and then clicks the chosen color. The stroke of any object which is currently selected will turn green, and the same color will be used for the outline of any element drawn in the future until the stroke color is manually changed.

[2] Well, yes, calligraphy is usually down with a fountain pen rather than a paintbrush, but consistency is the hobgoblin of simple souls, etc.

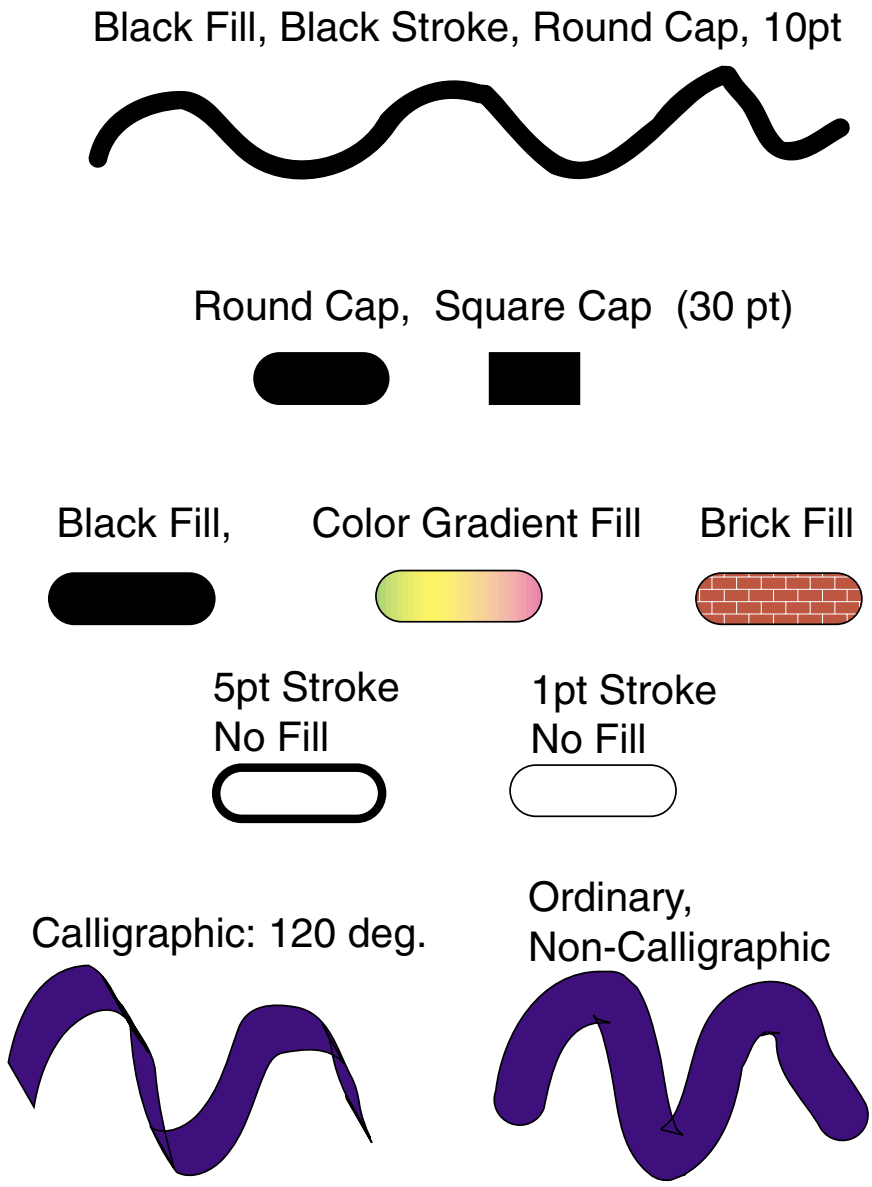


Figure 7.19: Illustrator brush tool variations. The brushstroke can be given both a "stroke", which outlines the swath of the brush, and a "fill". The stroke and fill can be of different colors or use textures like "brick" and various color gradients. The width of the brush can be varied from a 1 point swath to very large swaths.

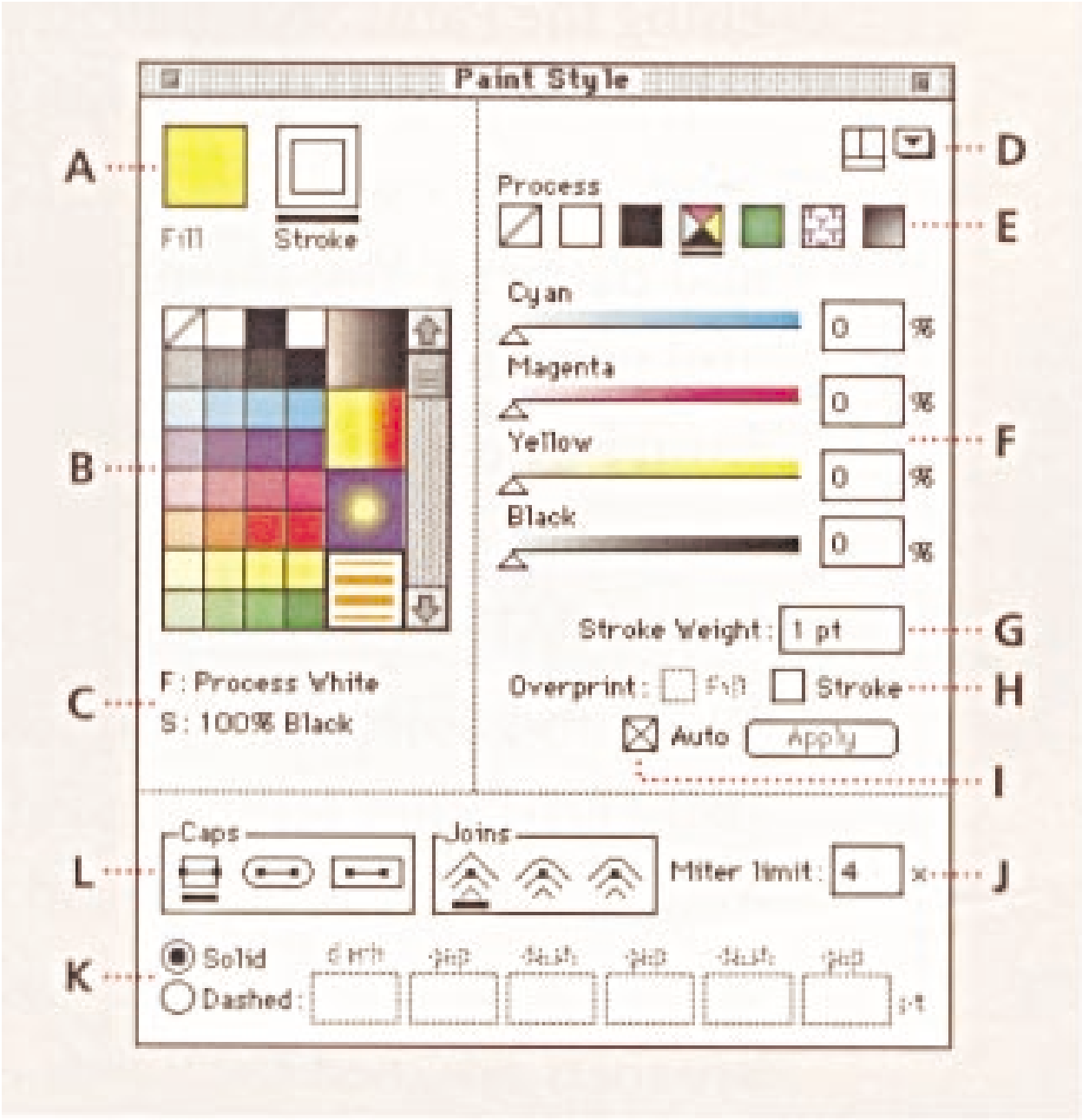


Figure 7.20: The paint palette from Adobe Illustrator, version 6. This is used to change the stroke and fill colors and also the characteristics of the stroke (line width, solid versus dashed, rounded curves versus "mitered" joins) as well as the pattern of the fill, which may include gradients of color or greyscale or complicated patterns like a brick wall, confetti, fish scales and so on.

On the right, labelled by letter "E" in the figure, are several boxes that access menus for custom colors (the four-color) box, named solid colors (solid colored box), patterns (a box filled with a pattern) and the rightmost box gives a menu of predefined gradients. Illustrator allows the artist to add his/her own creations to each of these menu lists; the procedure for creating new patterns is described in a later section.

The lines second from the bottom (label "L"). "Caps" refers to the end of a curve or line segment. These ends can be squared, rounded or square but extending a little beyond the control endpoint. "Joins" refers to what happens at sharp bends and corners: these can be rounded, square or "mitred".

A very useful capability is the freedom to change the linewidth (stroke width): type a new number (which can be a fraction) in the box labelled "G" where the number is the width in points (1/72 of an inch). Differentiating curves by thickness is very useful in black-and-white graphics.

The lower left of the paint palette (labelled "K" in the figure) allows one to specify a dash pattern. It is only necessary to type numbers into two of the boxes to the right of the "Dashed" button, but if one fills additional boxes, one can create a fairly elaborate dot-dash pattern, or even long dash-dot-short dash.

The fill of an object can be similarly altered by clicking the fill box in the upper left of the palette so that a little black bar appears underneath the box. For fills, one can not only change the color, but also specify textures and gradients of colors. (The stroke can be also be given a texture or pattern, too, but these are more restricted than for the fill and may not display or print clearly unless the stroke width is at least several points; gradients are not allowed for the stroke.)

The four color bars to the right of the color patches show the mixture of colors in whatever color swatch is currently active. One can manipulate the sliders on each color channel to define new colors which can then be added to the array of swatches and used throughout the rest of the document. (New shades can be permanently added to Illustrator by including them in the "startup" file, which allows the artist to specify all sorts of default preferences; otherwise, new colors are stored only with the document in which they are defined.)

7.16 Patterns

Illustrator unfortunately comes with a very limited suite of default patterns. Shading-by-diagonal lines is a common strategy in scientific graphics, but no such patterns are available. The "Waves" and "Confetti" patterns are sometimes useful, but little else, and the waves pattern is in a pale blue, suitable only for color graphics.

The reason Illustrator provides so few defaults is that it is easy to make a new pattern.

To make a fill pattern, the first step is to switch to the "artwork" view and then draw a small rectangle. (By double-clicking the square/rectangle tool, the square menu is popped up, which allows one to specify a standard size such as 0.25 inches square.) Set both the stroke and fill equal to "None" unless one wants to include the walls of the rectangle as part of the tiling pattern.

The next step is to draw the desired pattern within the square. To draw diagonal lines, for example, one can simply draw a line (of any chosen thickness and color) from one corner to another. Shift-click so as to select both the rectangle and the artwork within it. Then access the "Object" menu at the top of the screen and drag down to "Pattern".

A pattern creation menu will appear, listing existing patterns. Click on the "New" button and insert the desired pattern name, then "OK". The new pattern can then be used just like any other pattern in Illustrator.

Well, *almost*. New patterns are recognized only within the file that created them. One can make the pattern available to other figure files by opening both the figure with the desired pattern ("Source" file) and also the "Target". In the target file, choose File > Import Styles (that is, drag the File menu at the top of the screen down to Import Styles). Select the desired "Source" file and click "Import". All custom patterns, colors, and graph designs are then imported into the Target file.

Alternatively, Illustrator has a standard Startup file. One can make a backup of the default Startup file and then edit the Startup file by importing all the desired patterns, etc. (Important: one must create a bunch of rectangles (or other shapes) and apply each new pattern to one of the rectangles; unused patterns, it turns out, are *not* saved with the Startup file, thus causing it to revert to its original form.) Changes to the Startup file will then become available to all Illustrator files.

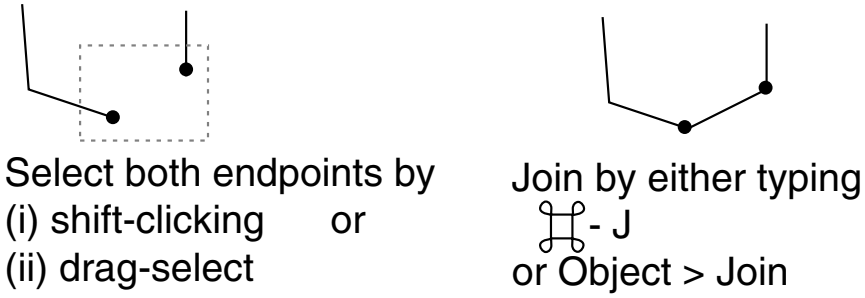


Figure 7.21: Adobe Illustrator's join command. To select the two endpoints to be joined, either hold down the shift key while clicking on both points, one after the other ("shift-click") or drag with the selection tool (solid arrow). A "selection marquee" (dotted rectangle) will appear; release the mouse when the marquee covers both endpoints. One can then join the two endpoints by typing "J" while holding down the Apple key, marked with both an apple icon and a funny symbol shown on the graph, or one move the mouse over "Object" on the menu bar at the top of the screen, drag down until "Join" is highlighted and then release the mouse. A line segment will be automatically drawn to connect the two endpoints.

7.17 Joins, Intentional and Inadverdent

Curves in Illustrator can be joined in three ways. The first is to use the "Join" command, which is accessed from the "Object" menu on the menu bar at the top of the screen or by Apple-J. One must first use the direct selection tool (hollow arrow) to select both endpoints that are to be joined, either by holding down the shift key while clicking both endpoints, or dragging with the hollow arrow to make a selection "marquee" rectangle. (Fig. 7.19.) A straight line will be drawn between the two endpoints unless the points are on top of one another. Note that only endpoints can be joined; an attempt to join a line to a corner of a rectangle will produce only an error message. (A line and a polygon can be effectively joined through the group command, Apple-G, however.)

Second, if the pen tool cursor is moved over an endpoint while the curve is selected, then the little mark below and to the right of the pen changes shape to indicate a state of "Over-the-Endpoint". If one then clicks and draws with the pen in the usual way, the resulting line segment or curve will be joined to the already existing curve. This feature was designed to make it easy to extend an existing curve.

Unfortunately, this feature sometimes leads to annoying accidental joins if one starts to draw while a pre-existing segment is still selected. In paper-and-pencil drawing, an element is automatically de-selected as soon as the pencil lifts from the paper. In a drawing program, the last element remains selected until it is manually de-selected. The "Deselect-All" command, Shift-Apple-A, is one of the most useful and frequently-used commands in Illustrator!

The final means of joining curves is to select two objects and then apply the Unite filter (from the Filter menu > Pathfinder > Unite). The two objects are merged into a single object with a stroke running around the exterior of the united object, but with stroked lines that are now on the interior erased.

7.18 Cutting Paths, Add/Delete Anchor Points & Converting Corner Points

The scissors tool (on the main tool palette) and the knife tool (on the plug-in tools menu) are used to cut paths. The scissors tool makes a point cut: two anchor points are added to a line segment at the point where the mouse is clicked. Both anchor points are open points; the line segment is broken at that point. If one point is moved, then this will open up a gap between the two initially coincident anchor points.

The knife tool has to be dragged across a boundary because it literally makes a slice in the interior of the object as well as cutting the side. Two coincident anchor points are placed at the point where the knife is dragged across a side of a polygon (or whatever), but these are connected through another anchor point which is placed where the dragging stops in the interior of the polygon or other object. By moving these anchor points slightly apart, we can open a fissure or crack in the object. (Fig. 7.20).

A third method to cut a path is to drag the Object heading on the menu bar at the top of screen down to the Apply Knife item. The selected object, if in the foreground, will then cut all objects that it overlaps as shown in Fig. 7.21. This is a very powerful device; cutting a circle with another circle generates a crescent and a lens simultaneously. One can thus cut very complicated shapes in a single command. The only complication is that the "knife" object is deleted — save a copy first if you need the knife later.

The add-anchor-point, delete-anchor-point and convert-anchor-point tools share the same box as the scissors tool. The implication is that these tools are not very important. In reality, novice artists will use them a lot. The functions of the add-anchor-point and delete-anchor-point tools, which have icons identical to the pen tool except for subscripts + or −, are self-explanatory.

The change corner-point tool actually toggles an anchor point between the status of a CORNER point, which is connected to its two neighbors by line segments that are straight at the corner point and meet at an angle, and a SMOOTH point where the slope of the curve is continuous at the anchor point. After the change-corner-point tool has been selected by clicking on its icon in the main toolbox palette, a small wedge-shaped cursor appears. When this becomes hollow, the cursor is within the active proximity of an anchor point. Clicking changes the status of the anchor point from smooth to corner or corner to smooth.

After a corner point has been changed to smooth, one must drag the direction points from the smooth point. If the direction points are dragged only a short distance, then the curve will be almost a corner point in the sense that the curve will turn rapidly from one slope to another as specified by the two direction lines, and then will be almost straight until near another anchor point. If the direction points are dragged far from the anchor point, then the curve will curve strongly even far from the anchor point, and the corner will be much more drastically rounded.

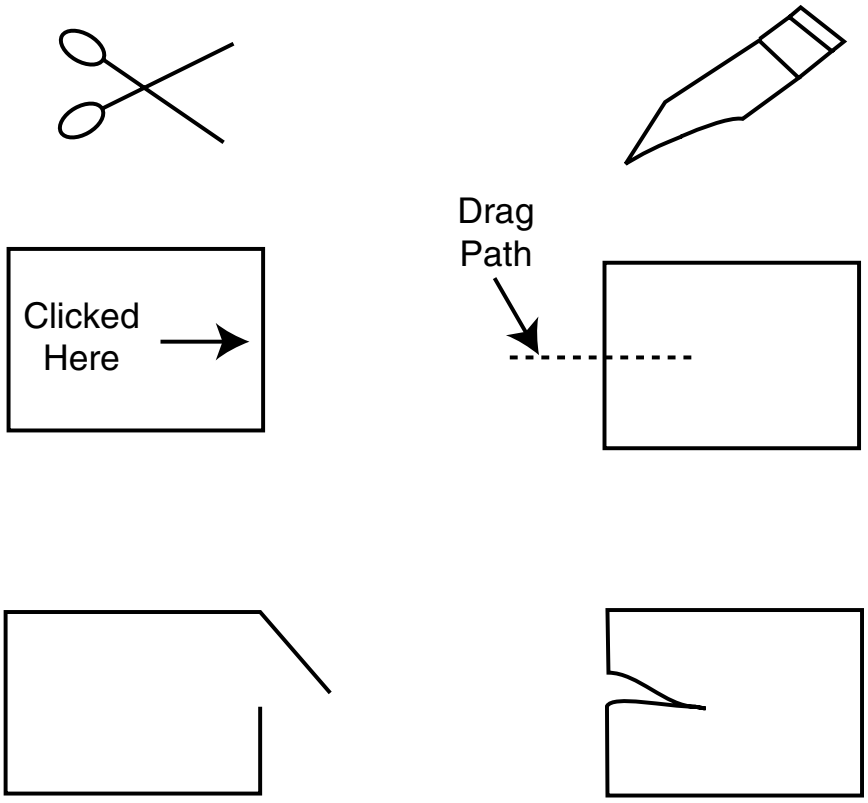


Figure 7.22: Adobe Illustrator's scissors tool (left column) and knife tool (right) are used to cut "paths", which is Illustrator-jargon for a curve or line segment. The icons for each tool are at the top. The rectangles at mid-level illustrate where the mouse is clicked (scissors) or dragged (knife) to make a cut. The scissors tool creates two coincident anchor points which are not connected; one can be moved to open up a gap as shown in the bottom left. The knife tool creates three anchor points: one on the interior of the object and two coincident points on the side of the object; none of these three new anchor points are endpoints. By moving the new anchor points on the side of the object, one can open up a crack as shown in the bottom right. It can be a little tricky to get the coincident anchor points apart, so one may have to experiment to get the crack or opening just right.

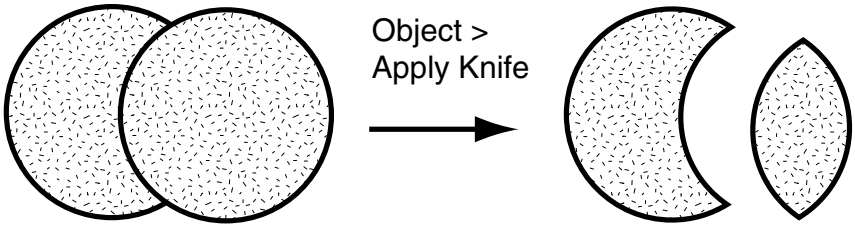


Figure 7.23: Adobe Illustrator's Object/Apply Knife allows one to cut an object using the outline (i. e., stroke) of the foremost object as a knife to cut objects behind it. Thus, if one selects the foreground circle of a pair of overlapping circles and then selects the Apply Knife option from the Object menu in the menu bar at the top of the screen, the "knife" circle is deleted while the circle behind it is cut into two pieces, which have been moved apart for clarity on the right.

7.19 Text Tricks

The Illustrator text tool, symbolized by the icon "T", allows one to enter text. Select the tool, then click where the text is to begin, and then begin typing text (including Return) as needed. Deselect the text box to end a given block.

Editing a block of text is mildly confusing because sometimes one wants to select the block as a graphic object (so as to move it or convert it to an ordinary graphics object) and sometimes one wants to modify only the text. The rule is: to select-for-moving, use the selection tool (solid arrow at top of palette); to select-for-editing, select the text tool ("T" icon) and then click on the text block.

Another mild complication is that it can be tricky to find the "trigger point" for a text block such that clicking near the "trigger" will select the block. Switching to the artwork view (Apple-E) makes it much easier since the trigger point appears as a little "x" in this view; Apple-Y toggles back to the artwork view.

It is easy to choose font, style, color and other text properties from the Type pull-down menu at the top of the screen.

Illustrator offers two additional text tools which are important because they are capabilities almost never found in scientific software such as Matlab. The "path type" tool allows one to attach text to a path so that it follows the undulations of a path. Thus, one can attach a curving line of text to a contour line even when the graph is so crowded with isolines that horizontal text won't fit.

The "area type" tool ("T" inside a closed curve) allows one to type text so that it is constrained to fit within a given shape such as an ellipse or a polygon. This, too, is a text capability which is rare in scientific software, but very helpful in fitting labels into a graph, as advocated by Tufte and most other graphic doyens.

A couple of text-changing options are often useful in scientific graphics. Type > Alignment > Center allows the text to be center-justified, which, for labelling, often looks better than the default left-justification. Type > Alignment pops up a dialog box; by first selecting a letter or letters and then typing in a number in the "baseline shift" box, one make the letters appear as subscripts or superscripts. The "baseline shift" must be negative for subscripts. A good choice is to set the shift equal to about 25%-40% of the type size, i. e., 3 to 5 points for 12-point type. The selected type won't actually shift on the screen until one hits the return key after entering the nonzero "baseline shift" number.

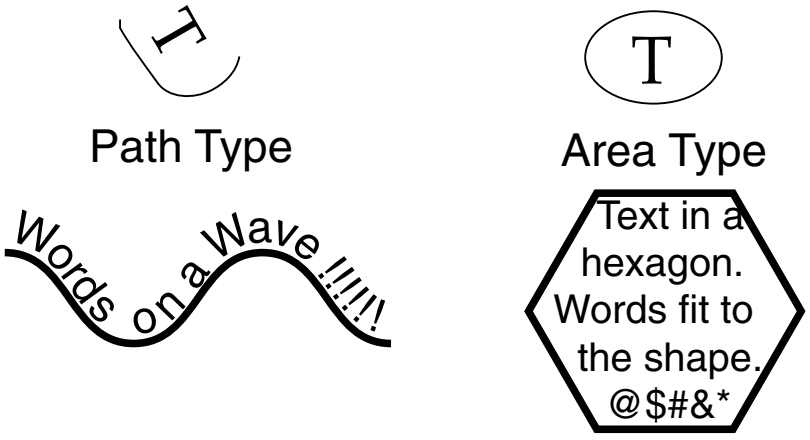


Figure 7.24: Schematic of the path-type and area-type tools; these are accessed by dragging the normal text tool (“T”) icon to the right.

7.20 Shape Drawing Tools: Ellipses, Rectangles, Stars, Etc.

Each of the shape drawing tools in Illustrator — oval and rectangle tools on the main tool palette, and the spiral, star and polygon tools on the "plug-in" tools palette — offers a number of options.

Oval tool — select the tool and then drag to draw an ellipse. (It is misnamed the "oval" tool in the Illustrator literature; it always draws an ellipse or circle.)

1. Circles are drawn by holding down the SHIFT key while dragging.
2. Circles and ellipses of a desired size may be drawn by clicking once at the spot where the ellipse is to appear on the drawing, which pops up a menu that allows one to type in numbers for the two axes of the ellipse.
3. By default, dragging generates an ellipse with an edge at the start of the drag; by double-clicking the oval tool in the palette or by holding down OPTION while dragging, one can draw from the center so the start of the drag is the geometric center of the ellipse.

The ellipse is always drawn with horizontal and vertical axes, but it can always be rotated after creation.

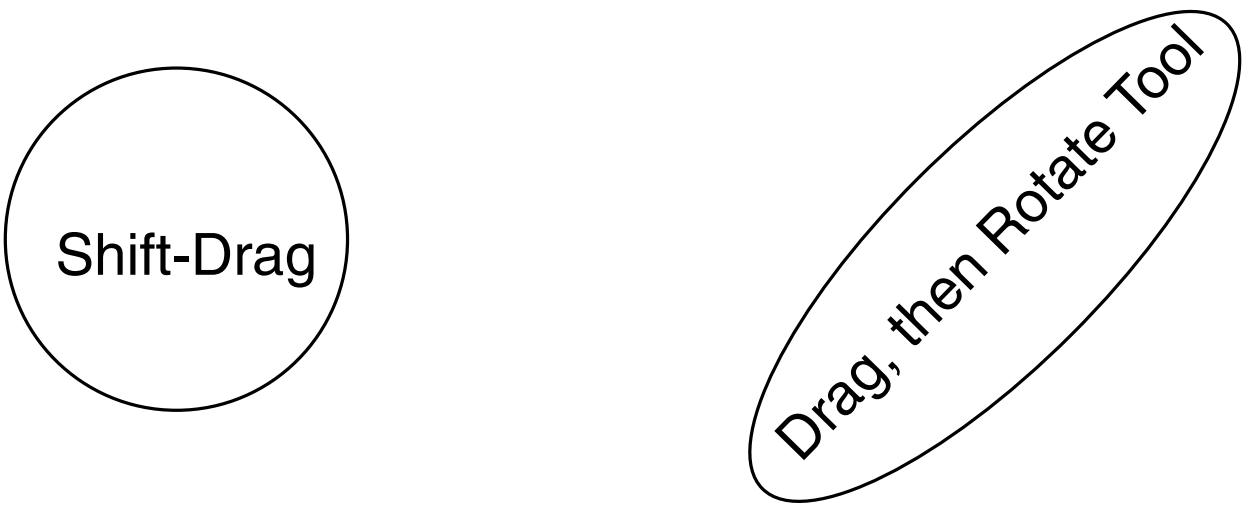


Figure 7.25: Top: capabilities of the (misnamed) "oval" tool in Illustrator. After the tool is selected, one can draw by dragging either from the side (if the tool is selected by a single click) or from the center (if the tool is double-clicked; a plus sign will appear in the icon). Shift-dragging generates a circle. An ellipse is always generated with semimajor axis parallel to either the vertical or horizontal axis, no matter how one drags, but an ellipse with slanting axes can always be generated by applying the rotation tool.

Although the shape drawing tools are DIRECTLY capable of drawing only very simple, regular shapes, it is possible to build very complicated shapes by applying other tools to the regular shapes. For example, the scissors tool can be used to draw circular arcs, even though this is not a capability of the oval tool *per se*.

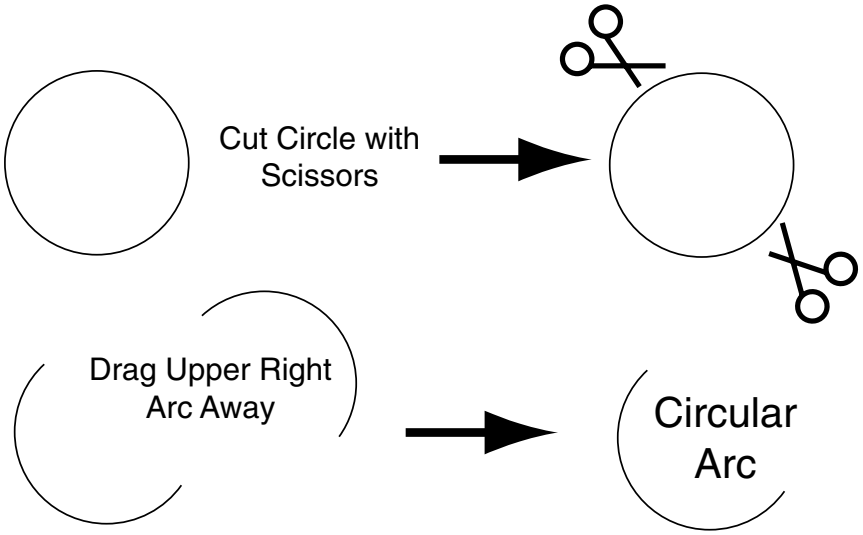


Figure 7.26: The oval tool cannot draw a circular or elliptical arc directly. However, it is easy to draw a circle, cut it in two places using the scissors tool, and then drag away the excess to obtain a circular or elliptical arc.

By using other tools, more complicated figures can be drawn from circles. Circular arcs can be duplicated, reflected, rotated and assembled to make highly structured figures.

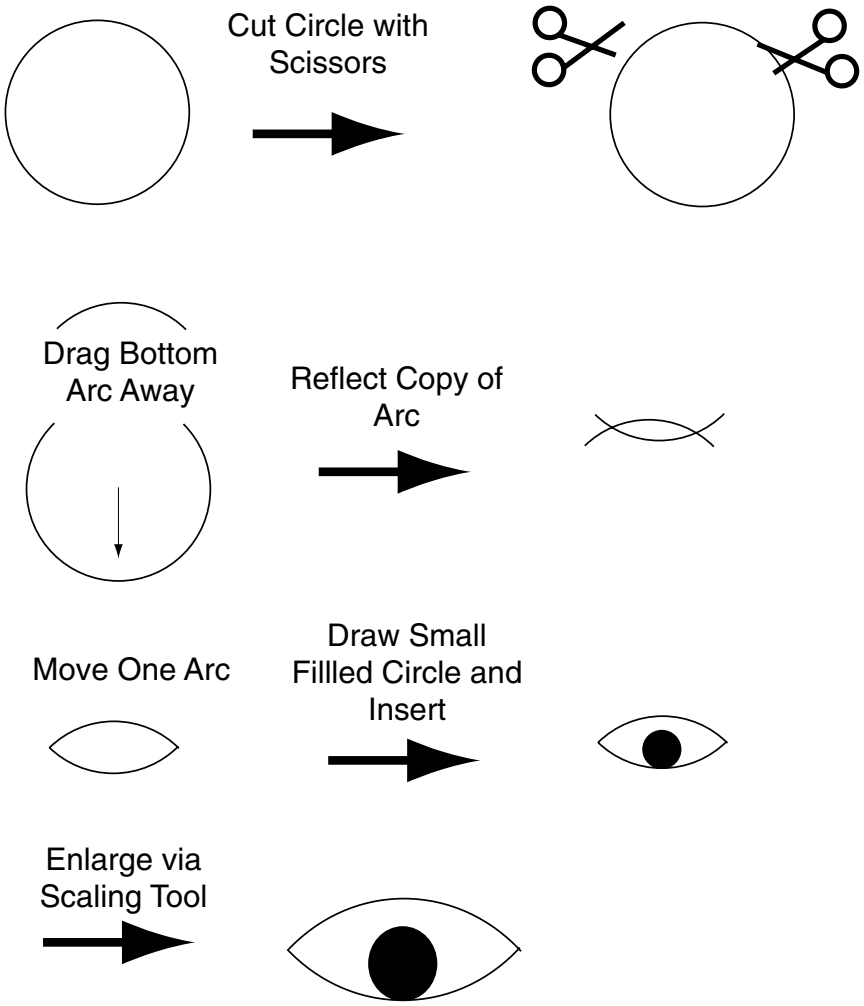


Figure 7.27: An eye can be drawn by using the circle tool twice. A large hollow circle is cut with the scissors tool to make a circular arc. The reflect tool contains a "Copy" button which will apply the tool to a copy of the arc. One arc can then be moved by select-and-drag to complete the eyesocket. The eyeball is simply a filled circle. After the objects in the eye are grouped (via selecting them all through shift-click and then the Apple-G keyboard command), the scale tool will resize the eye to any desired height and width. Using one circle to cut another with the Object > Apply Knife option will also generate a lens-shape from output of the circle tool and is probably faster than applying the scissors tool twice.

Rectangle tool — click in the toolbox palette to select the tool and then drag to draw a rectangle.

1. To draw a square, hold down SHIFT while dragging.
2. Click-instead-of-drag to pop-up a dialog box; enter numbers for the sides of the rectangle, which will appear at the point of the click.
3. By default, dragging draws from a corner, but double-clicking the tool or OPTION-dragging will draw from the center, that is, the center of the rectangle will be at the point in the figure where dragging begins.
4. The pop-up menu offers the option to round the corners of the square by circular arcs of a user-specified radius.

The rectangle is always drawn with horizontal and vertical axes, but it can be rotated after creation.

For all these tools, the object is drawn with the current stroke and fill, which can be altered in paintstyle palette.

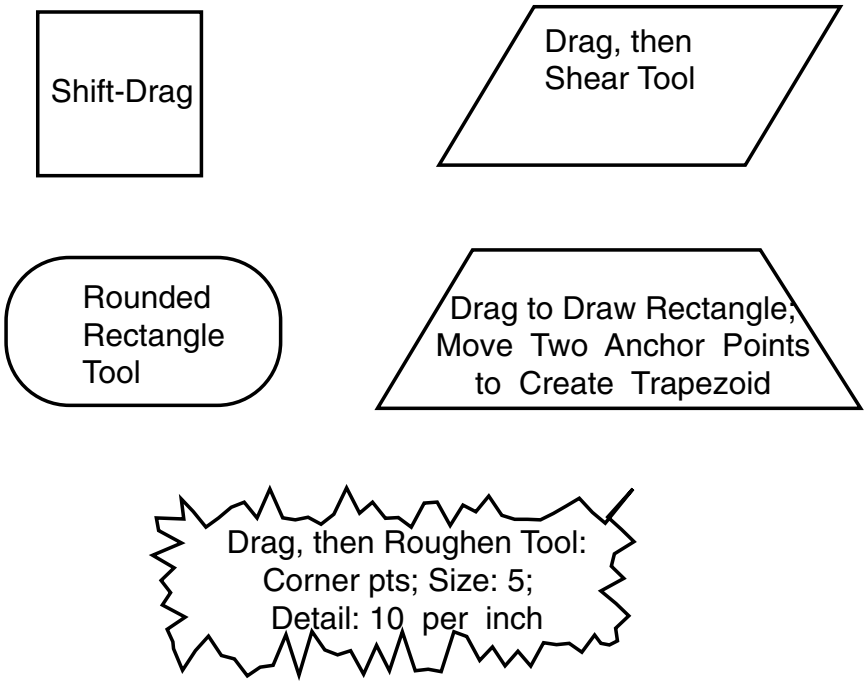


Figure 7.28: The rectangle tool is similar to the circle tool except that it has a third option, which is to generate a square or rectangle with rounded corners. By double-clicking on the artboard (not the tool icon), one can pop-up a menu that allows one to specify the width and height of the rectangle and the radius of the circular arcs for the corners. It is not possible to generate a rhombus by the rectangle tool all by itself, but one can draw a rhombus by applying the shear tool to the result of the rectangle tool. Similarly, a trapezoid can drawn by moving one anchor point of the rhombus using the direct selection tool, or two anchor points of a rectangle. By applying filters to the stroke or fill, one can generate very complicated figures as illustrated by applying the roughen tool to a rectangle.

Similarly, stars, spirals and polygons can be drawn by select-tool-and-click-and-drag. The polygon tool is limited to "regular" polygons with an integral number n of equal sides, fit within an inscribed circle of radius r. By rotating, rounding corners, and applying other filters such as "Zig-zag" to the polygon, however, one can make a great diversity of shapes beginning with a regular polygon (Fig. 7.27).

The pop-up menu allows one to specify the number of points of the star, the number of segments of the spiral and the number of sides of the polygon along with other information such as the the inner and outer radii of the star, radius and decay percentage and clockwise/counterclockwise orientation of the spiral, and radius of the polygon. Draw-by-dragging allows one to experiment with the size; the objects can be all be rotated by dragging around the center of the object.

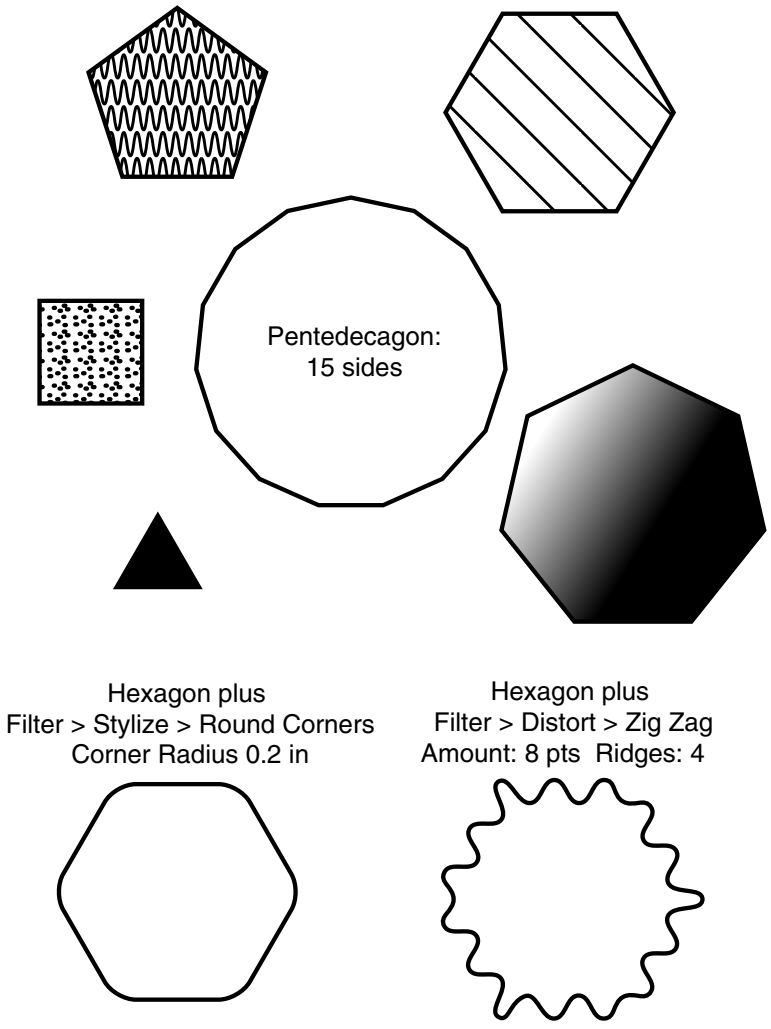


Figure 7.29: The polygon tool can draw polygons of an arbitrary number of sizes. The fill patterns and stroke can be varied as for any other graphic object. Filters can modify the polygons into more exotic shapes. The "round corners" tool and "zig-zag", which makes stroke lines wavy with either smooth oscillations or jagged (corner point) oscillations.

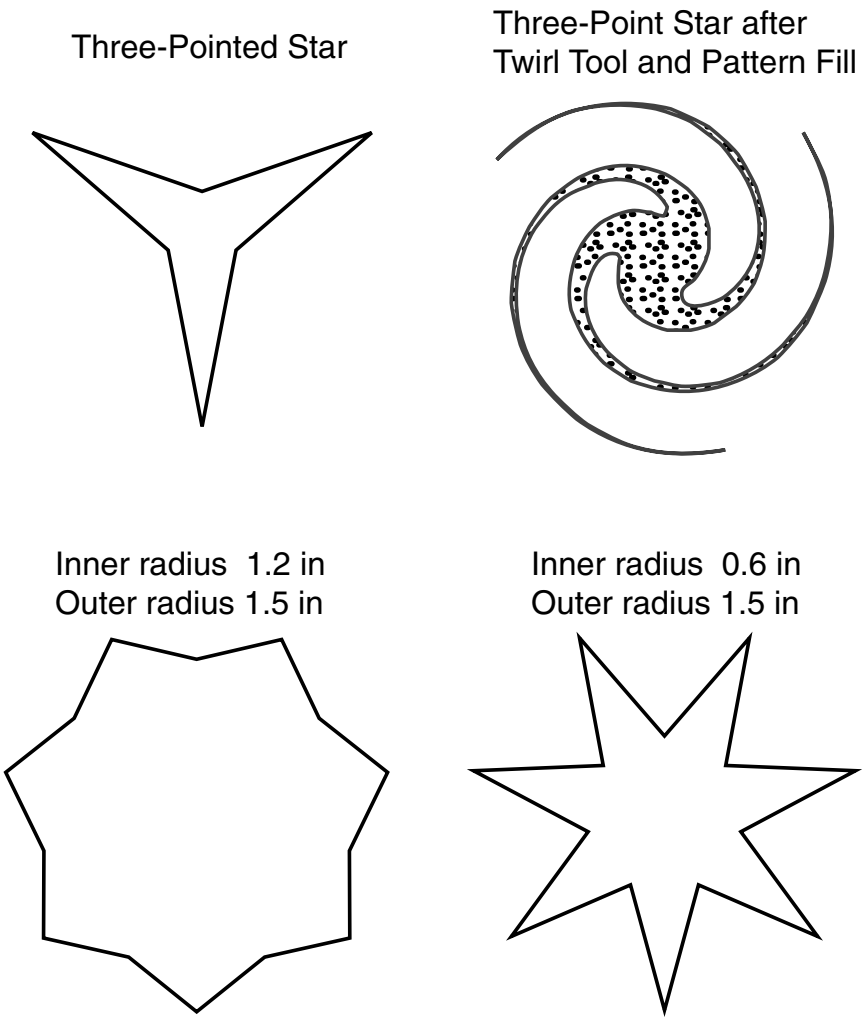


Figure 7.30: The star tool can vary the number of points from 3 to infinity. The shape can also be manipulated by specifying a large or small inner radius. The star tool is on the plug-in tool palette. Its menu can be accessed by double-clicking on the artboard (where the drawing appears) after selecting the tool.

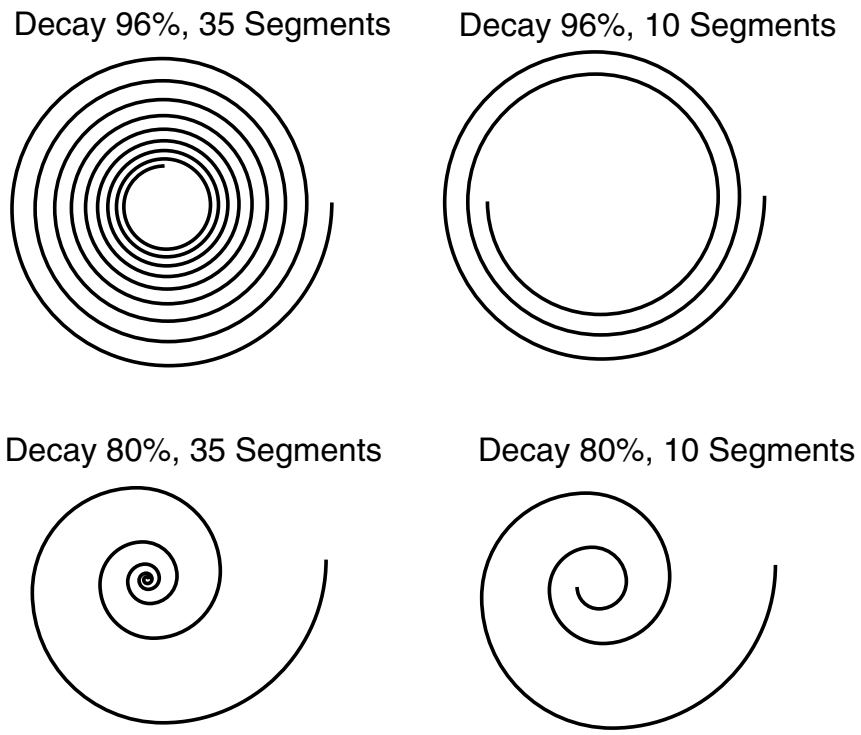


Figure 7.31: The objects drawn by the spiral tool depend on three parameters. One is the radius of the circle which circumscribes the spiral; this is the same for all four spirals shown. The second is the "decay rate"; when this is close to 100%, the curves of the spiral are almost on top of one another. The third is the number of "segments"; when this is small, the curve only circles the origin a few times; when the number of segments is large, the spiral circles the origin many times. These four spirals all rotate clockwise as one moves closer to the center, but a counterclockwise orientation can be specified by clicking the appropriate button in the dialog box.

7.21 Transformation: Rotate, Reflect, Shear, Scale

All these tools require a center point as well as a magnitude plus additional information for some tools. All transformations require that the object to be modified is selected, using the usual selection tool (black arrow), before selecting the transformation tool.

To use the center of the figure as the axis of rotation or reflection, there are two ways to proceed:

1. Dialog box: double-click the tool. When the dialog box appears, specify the angle.
2. Drag method: click the tool and then drag-without-clicking.

To choose another point as the axis of rotation or reflection, there are two ways:

1. Dialog box: single-click the tool, hold down the Option key and click on the new center point (the cross-hair cursor will change to a black wedge). Enter an angle in the dialog box.
2. Drag method: single-click the tool. Click with the crosshair cursor to set a new center point and then drag to transform.

One useful generic feature is that the dialog boxes always contains a "Copy" button. If this is pushed, the transform tool will generate a copy of the selected object and transform the copy. A common way to generate symmetric objects, for example, is to draw *half* the object, reflect-a-copy, and then move the two halves together. The group operation (if a line down the center is desired) or the Unite filter (to erase the line segment down the joining axis) can then permanently weld the halves together as in Fig. 7.30.

All transforms can be repeated by typing Apple-D; if the "Copy" button is pressed with a rotation angle of say, thirty degrees, then the transform is applied to the copy-of-the-copy. To draw a flower, for example, one can draw a single petal and then rotate-with-copy four times to generate all five petals at the proper angles around the center of the flower, which should be used as the rotation point.

The dialog box is especially useful for scientists because it allows one to specify precise numerical angles and, for the reflect tool, whether one wants to reflect about a vertical or horizontal axis. The reflection tool can reflect with respect to a tilted axis, but this is a capability one won't use much.

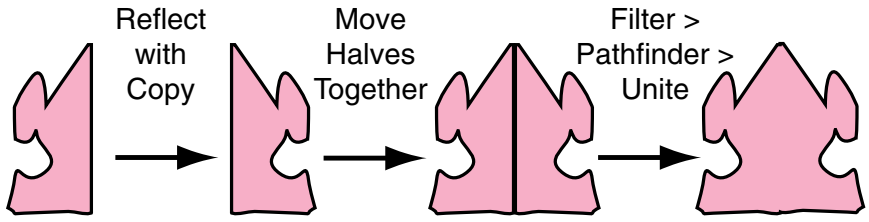


Figure 7.32: The Reflect-a-Copy option of the Reflect tool makes it very easy to create figures which are symmetric about a central vertical axis (or other axis).

The shear tool is for very useful for converting a rectangle into a rhombus, such as the side of a cube or parallelpiped shown in perspective.

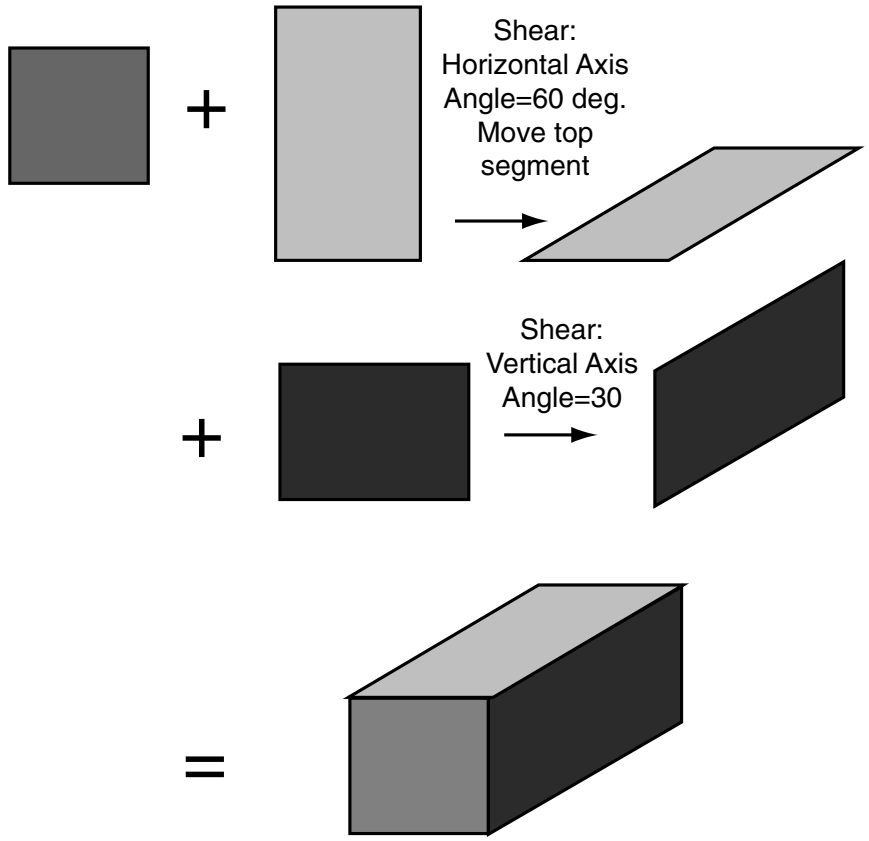


Figure 7.33: The shear tool, combined with the rectangle tool, is a quick way to draw a cube in perspective.

The scale tool can dilate an object, but one can specify different expansion or contraction factors for the horizontal and vertical dimensions. One can also scale-by-dragging to change the proportions in rather interesting ways. In particular, one can choose to dilate about a point which is not the center of the object.

The Arrange menu at the top of the screen offers an alternative way to scale through Arrange > Transform Each, followed by entry of the horizontal and vertical scaling percentages in the dialog box. When the scaling tool is applied to a *group* of objects, each object is rescaled around its own center point. If one has a grouped object that is a circle surrounded by six circles all touching its nearest neighbor at the rim, then the scale-by-Transform-Each will shrink each circle in place so that the circles no longer touch (Fig. 7.32). In contrast, the scale tool will shrink the seven grouped circles about their common center so that the circles are still in contact after contraction; the centers of the six outer circles have shifted towards the interior of the central disk. Transforming an entire group is thus rather tricky, and one must be careful to choose the proper scaling.

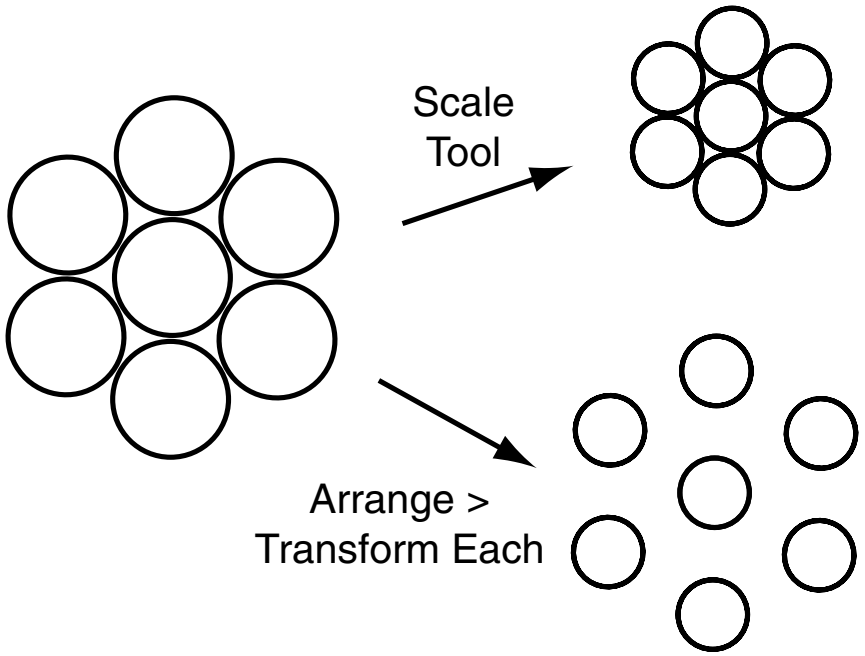


Figure 7.34: Two different ways to scale ("dilate") a group of objects with rather different results.

The "blend" tool is really a "morphing" tool: it is applied to two objects, and generates a set of objects between them which interpolate, in a uniform sequence, the shapes and colors of the outermost pair. The user can choose the number of intermediate objects. Although this is great for artists, it isn't often used in science.

7.22 Isometric Drawing and the Third Dimension

Fig. 7.48 shows isometric drawing. The best way to employ an isometric grid is to draw it once-and-for-all as a template file. This consists of the isometric grid (on one

layer) and a second blank layer. This file can be then be opened, saved under a different name so as to preserve the template file unchanged, and then edited by drawing on top of the isometric grid. When the drawing is complete, it is a one-step operation to delete the grid, leaving only the pseudo-perspective drawing.

The reason for the adjective "pseudo-perspective" is that in a true perspective drawing, objects at the back of the field of view are smaller than same-size objects in the foreground. Isometric drawing sacrifices this aspect of perspective drawing by drawing all objects at the same size. This enormously simplifies the drawing; engineers are not trying to please art patrons but merely give machinists a visual representation of the object they must build. The equal-size property of isometric drawings is actually an advantage in engineering because measurements can be taken directly from the drawing without the need to correct for the continously-varying scale of a true perspective drawing.

For serious design work, one should use a three-dimensional CAD-CAM system. To avoid the steep learning curve of such software, however, isometric drawing may be preferable if one has a small number of quasi-three-dimensional shapes to sketch.

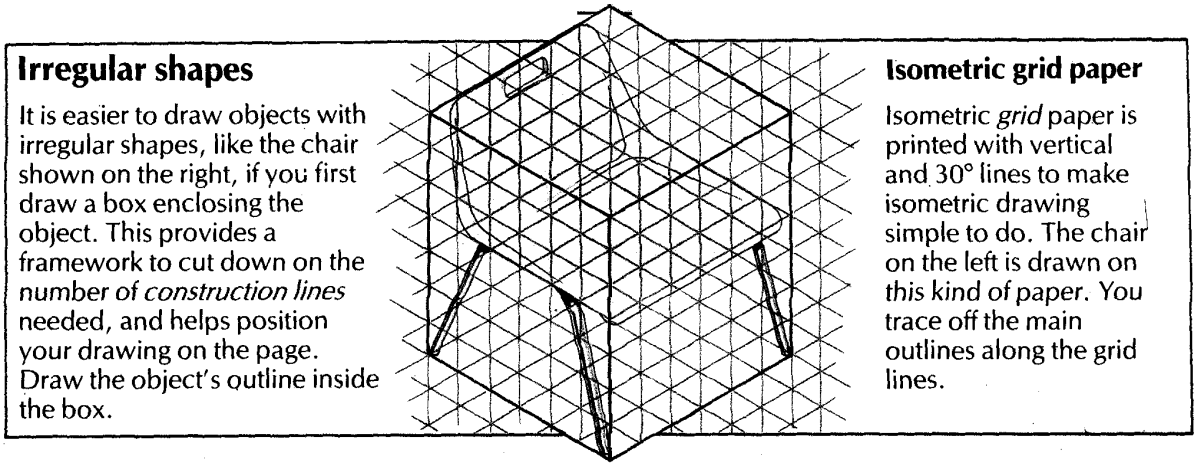


Figure 7.35: Drawing of a chair on an isometric grid. Scanned from pg. 22 of *Technical Drawing: An Usborne Guide* Usborne Publishing, written by Susan Peach, published 1987

1. isometric grid is good use of layering; the grid layer can be deleted later
2. isometric drawing is popular for engineering and CAD/CAM because "all the sides of the object are drawn at their true length, so you can take measurements from the finished drawing".
3. isometric is FAKE perspective, which reminds us that engineers don't need to be artists; "faking it" is usually sufficient.

7.23 Filters, Styles and All That

Illustrator allows one to modify curves through a variety of operations that we will collectively call "Filters". These filters are so numerous that only a professional artist can learn them all, and many are useful only in artistic drawing. However, some filters can be helpful in scientific drawings.

Some exceptions and possible engineering uses are listed below.

1. Add Arrowheads . This is helpful to generate arrow-headed coordinate axes, arrows that point from labels to the labelled object and in schematics of velocity fields (quiver plot). The Filter > Stylize > Add Arrowheads command pops up a menu of 27 different choices. One can further specify from the menu arrow-at-end, arrow-at-beginning, or arrowheads-at-both-ends. This function works with both line segments (pen tool) and with curves draw by the freehand or pen tools.

2. Zizag filter. This makes WAVY lines from straight lines. The waves can either be rounded or jagged. Hint: the arrows look best when the line segment is horizontal or vertical when Zigzag is applied; one can then rotate the wavy line to any desired angle.

3. Round Corners. This adds rounded corners to any object with corners. When invoked by Filter > Stylize > Round Corners, a dialog box appears which allows one to specify the radius of the circular arcs that will placed at each corner point. One can always round the corners manually, but it is tricky to round each corner of a polygon by exactly the same amount; the Round Corners filter automates this process.

4. Free distort. This places the object inside an imaginary rectangle; a menu allows one to freely move the four corners of the parallelpiped, and then distorts the figure inside the box to follow the movements of the corners. Filters > Distort > Free Distort.

5. Roughen. If this filter is applied repeatedly with a variety of scales and sizes, a smooth surface can be distorted into an approximation of a fractal, such as a schematic coastline.

6. Unite. This allows one to join two objects so that boundary lines which are on the interior of the united object are automatically erased. This is good for drawing complicated shapes in schematic diagrams. (Filter > Pathfinder > Unite.)

7. Minus Front. This filter allows one to create a hole in an object so as to create an object that a mathematician would call "multiply-connected" and Illustrator calls a "compound path". The hole object is dragged in front of the larger object, both are selected using shift-click with the ordinary selection tool active, and then the filter is applied. The "hole" is automatically destroyed, so it is a good idea to make a copy if the hole is needed. The hole is transparent, so the hole can be filled by dragging an object with a fill pattern in front of the hole and then sending the object to the back using the Arrange menu. (Filter > Pathfinder > Minus Front.)

Fig. 7.33 shows that a wide variety of filters are available under the Distort submenu. The Distort filters operate by shifting the anchor points in various patterns.

The Pathfinder filters allow one to make complicated shapes from pairs or groups of simpler shapes. There are far too many variants to discuss in detail, but Fig. 7.34 shows the variety available.

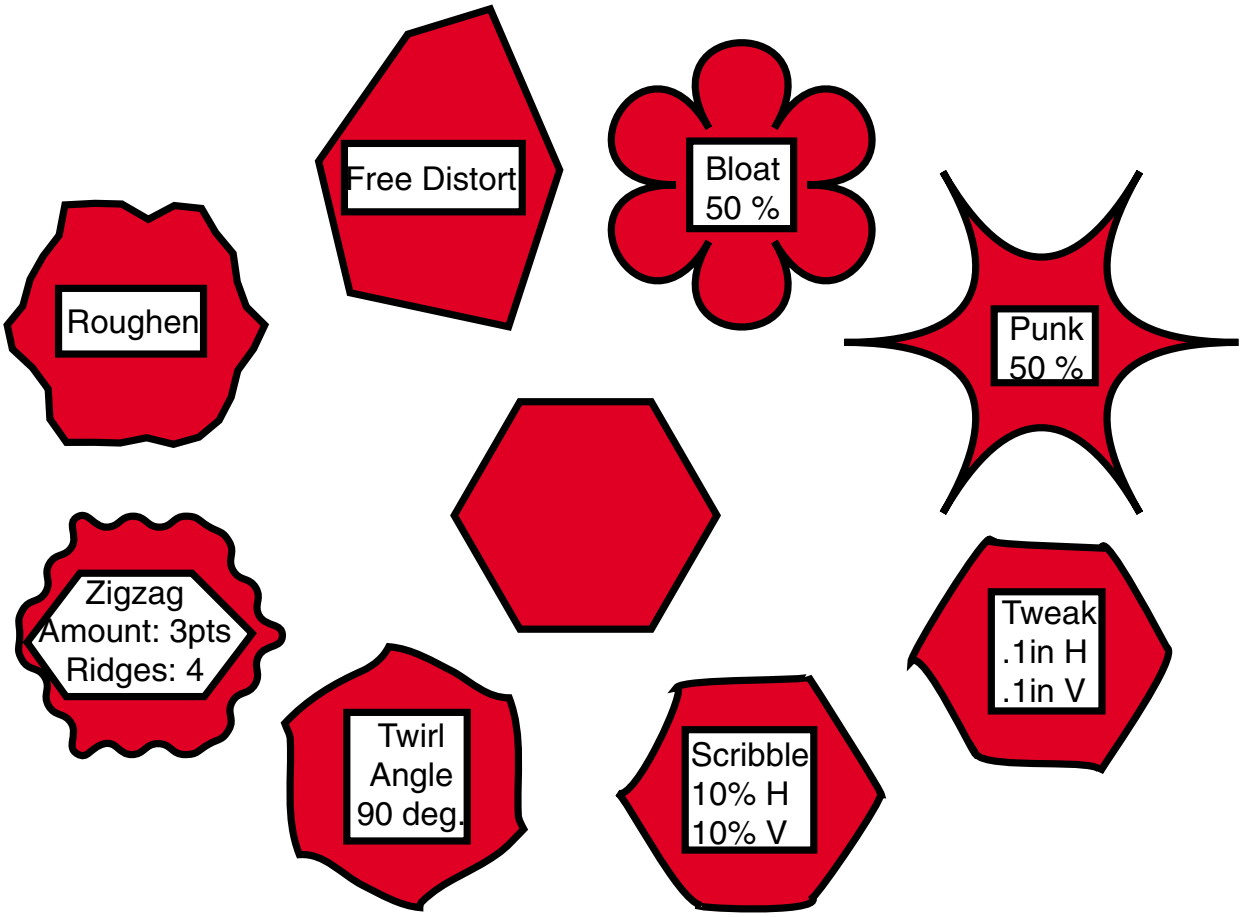


Figure 7.36: The effects of all the Distort filters in Illustrator version 6 on a hexagon. Many of the filters can user-adjustable parameters as indicated in the box. The "Bloat" filter curves the object outward from its anchor points (in this case); "Punk" produces an inward curvature. "Scribble" and "Tweak" move the anchor points themselves — randomly in "Scribble" and by the precise amounts specified in the dialog box in "Tweak". The "Twirl" filter is similar to the Twirl tool in that the inner parts of the figures are rotated more than the outer parts; the maximum rotation angle is specified in the filter dialog box. The "Zigzag" filter can generate waves with corner points as well as smooth points. The "Roughen" filter adds anchor points at small random distances from the original boundaries of the object.

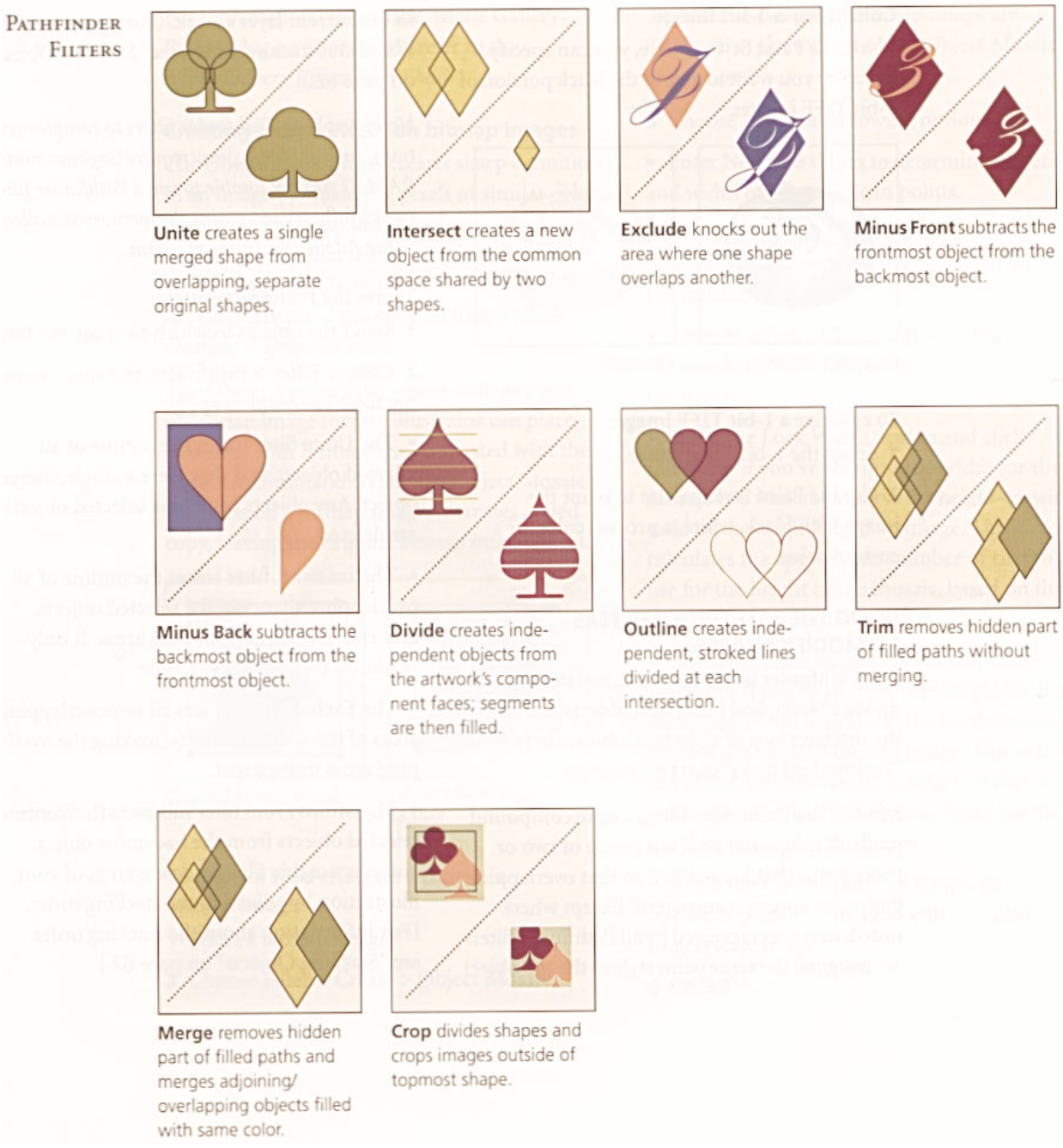


Figure 7.37: The Pathfinder menus. The Unite and Minus-Front are especially useful.

An engineer rarely needs to draw the sort of artsy shapes shown in the illustration of the Pathfinder filters. However, filters and other tools are very valuable in many kinds of scientific line drawing. The Feynman diagrams in Fig. 7.35 are an example.

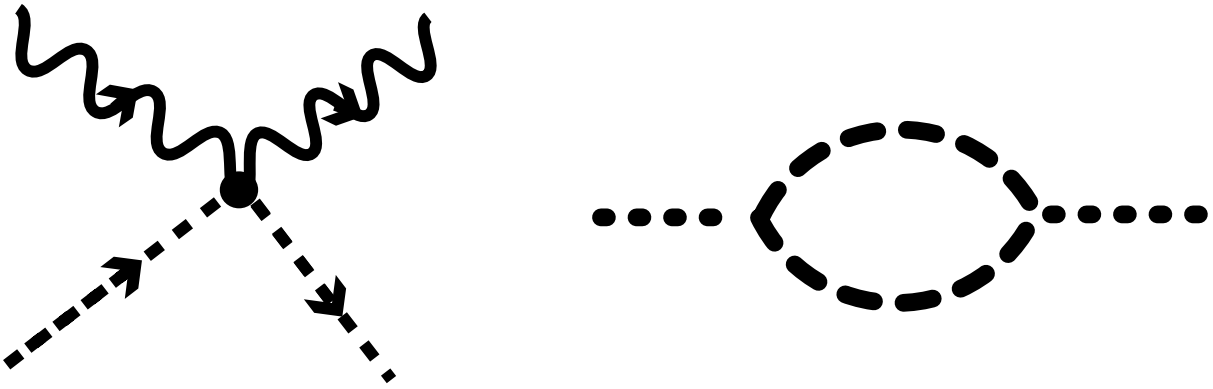


Figure 7.38: Feynman diagrams are essential tools in particle physics. By convention, different types of particles are represented by different types of lines: solid, dashed, dotted, and wavy. Feynman diagrams are simple enough to be drawn in Matlab (after creating some specialized drawing functions for wavy lines, arrowheads, etc.), but it is easier to draw them in Illustrator. The Zigzag filter created the waves in the wavy line; specifying dash and gaps in the bottom right of the Paint Style palette generated the dotted and dashed lines. The lens was created by cutting one circle with another using Object > Apply Knife and discarding unnecessary pieces. The arrowheads were added by drawing short line segments and then Filters > Stylize > Add Arrowheads. The arrowheads were rotated into place by using the rotation tool: the back of the arrow was dragged onto the line it was intended to adorn, then Rotation tool was clicked, then the back of the arrow was clicked with the crosshair cursor (to specify that point as the axis of rotation) and then the front of the arrowhead was rotated by dragging with the black wedge cursor until the arrowhead was aligned with the wavy line.

7.24 Pictographs alias Visual Tables

Pictographs, which are figures in which copies of a single, simple shape are repeated to indicate quantities, are uncommon in scientific graphics. Therefore, Matlab and most other scientific software lacks pictograph-making subroutines. Of course, pictographs are rarely used in scientific graphics in part because there are no easy ways to make them using standard scientific software!

Illustrator draws pictographs as follows.

Step One: Make a bar graph.

1. Click the graph tool at the bottom left of the main palette. Drag it diagonally across the screen to create a rectangle to contain the graph.

2. Graph creation automatically pops up a spreadsheet for data entry. Labels and numbers can be typed into the columns of the spreadsheet. (Numerical labels, such as a year, should be typed between quotes). As an alternative to one-at-a-time typing, an array of numbers can be copied from the Matlab screen as a row vector and pasted directly into a column of the spreadsheet. To edit the data, use

Object > Graph > Data where "Object" is a pull-down menu on the menu bar at the top of the screen, "Graph" is the bottom of the Object menu, and "Data" is an item in the "Graph" menu.

3. By default, Illustrator makes a "grouped column" chart and puts the legend on the side; both can be changed through Object > Graph > Style.

Step Two: Make a pictograph icon.

1. Draw the icon.
2. Draw a rectangle around the icon. Set the stroke and fill of the rectangle to "None".
3. Select the rectangle and then do Arrange > Send to Back where "Arrange" is a pull-down menu at the top of the screen.
4. Drag the selection marquee over both the rectangle and the icon within it and choose Object > Graphs > Design.
5. Click New.
6. Enter the name of the new icon, then click OK.

In Illustrator parlance, a pictograph icon is a "graph design". Illustrator identifies a possible "graph design" by looking inside the backmost rectangle; hence, we used the "Send to Back" command on the rectangle.

Step Three: Apply the pictograph to the graph.

1. Using the GROUP selection tool, select all the columns (and if there is a legend, the corresponding legend) using shift-click until all the columns in a given group are highlighted.
2. Objects > Graphs > Column.

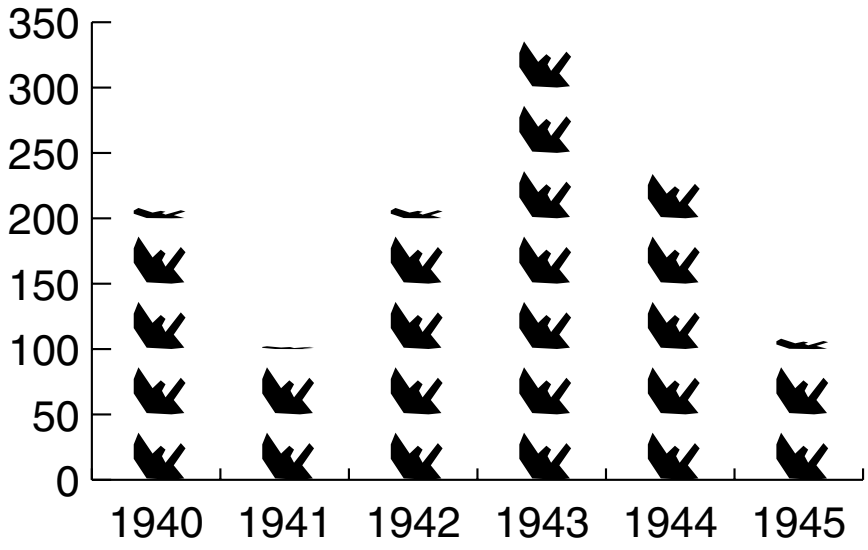


Figure 7.39: Pictograph of sunk allied merchant years during World War II (with made-up numbers!). Each icon represents fifty vessels sunk in a given year by German U-boats. The flattened icons represent a fraction of fifty ships.

3. In the dialog box, choose “repeating” in the column design box, to specify the quantity associated with each icon, such as fifty ships per ship icon. Fill in the box that will appear: “Each design represents” with the number of units per icon.

4. Group-select the second set of columns (for a second variable) by shift-clicking.

5. As for the first column, Objects > Graphs > Column, then choose “repeating”, and finally fill in the number.

6. When the graph is done, delete the rectangles-with-icons that were used to define the “graph design”.

Warning: it is easy to select only subelements of a column, so one may have to use the group selection marquee to select an entire column, as necessary to change it from a bar to a cluster of icons.

Illustrator saves the pictograph icons in its internal file format as part of the “style” of the graph. One may recycle icons by using the “Import Styles” command when making a new graph; this will allow the new figure to use pictograph icons created in an earlier figure.

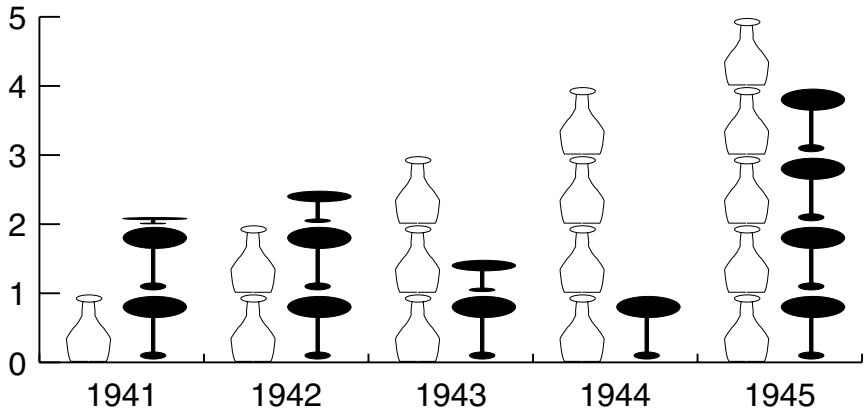


Figure 7.40: Two column pictograph. For each year, labelled at the bottom, the left column (milk bottle) icons depicts the consumption of milk and the right column (martini glass) the consumption of alcohol.

7.25 Image-processing in Science

When a photograph is to be used in a scientific or engineering publication, it oftens need editing. Broadly speaking, photo-editing can be grouped into three categories:

1. Extraneous details
2. Defects (fixing deficiencies of focus, color, etc.)
3. Overlay

Photographs often contain extraneous details. For example, a photo of the Michigan solar-powered car ("Sunracer") will likely include road surface, street signs, sky and a variety of other irrelevant features. Painting tools are very valuable in eliminating unnecessary details.

One strategy is to use the "lasso" selection tool to choose the interior of an irregularly-shaped object, such as the automobile, and then crop everything not selected. This is a particularly good way to show only the essentials without misleading because the irregular shape of the photograph shows that the rest of the image has been cut away.

An alternative is "cloning". Unobstrusive bits of background can be duplicated and then placed over background features that are brightly colored or otherwise eye-catching but irrelevant. However, this can be controversial because the photograph has been edited, and the editing is hidden.

Some years ago, a newspaper ran a photograph of a reporter who had just won the Pulitzer Prize in journalism. The photograph had been taken in the reporter's office in great haste. When the photo was developed, the editors discovered to their horror that the most prominent feature was a bright red aluminum can of Coke Classic on the edge of the reporter's desk. The newspaper therefore airbrushed the can away, replacing the hole with "cloned" background. This caused a major controversy when the hasty retouching was noticed by a rival publication. The Pulitzer Prize is awarded for *accurate, truthful* reporting — but the picture of the prize-winner was neither accurate nor truthful. On the other hand, what is the relevance of a Coke can to a journalism prize?

In spite of the "Coke can" controversy, elimination of extraneous details from photographs is widely practiced in science and engineering. The use of filters to sharpen blurry images is also legitimate. However, it is important to clearly state in the caption and the text what filters have been applied to the raw image, especially to the essential, scientifically-relevant parts of the image.

Photographs often contain defects that can to some extent be fixed in Photoshop or other image-processing program. The "sharpen" and "blur" tools (really just Diffusion and Anti-Diffusion filters) can partially compensate for an image that is a little bit out of focus or otherwise has poorly defined edges. Scientific images often have poor contrast: Voyager photos of Neptune for example, are basically green-on-green. By increasing the contrast, many features become visible such as the large anticyclonic storm that covers much of the planet's surface. To enhance contrast, the computer makes pale green very, very light and a slightly darker green becomes very, very dark. A Photoshop user can play with the "sharpen", "blur" and contrast controls to obtain a processed image that shows the greatest amount of detail.

Scanned images are often contaminated by dust. These stray pixels of dark on light or the reverse can be deleted by hand, but Photoshop has a filter that automatically removes isolated pixels that differ by a user-set tolerance from neighboring pixels — a sort of electronic lint-cleaner.

A third use of painting tools is to overlay information on the photograph. Fig. 7.38 is a good example. The satellite cloud photograph is almost incomprehensible by itself. However, radiosonde observations have identified the warm front and cold front. By using drawing tools, one can superimpose the fronts and coastlines on the cloud photograph so that one can discern what features of the clouds are correlated with the fronts.

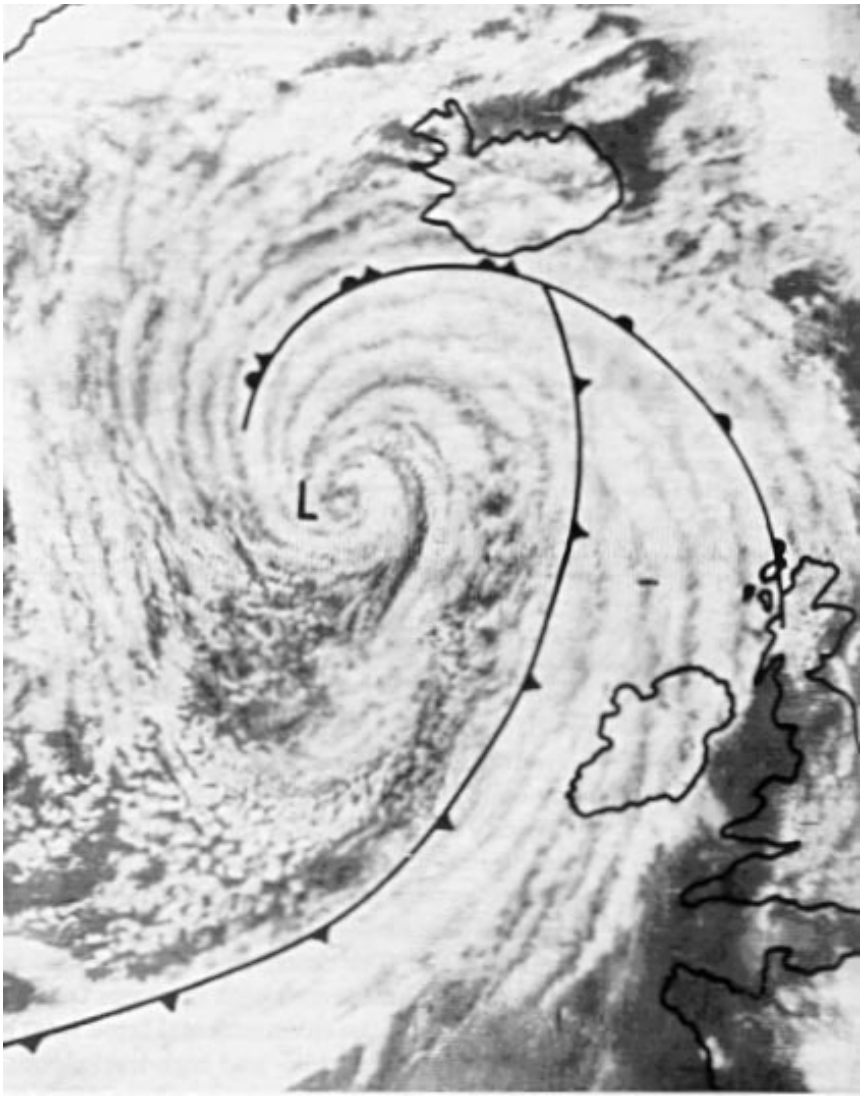


Figure 7.41: Scanned from *Weather Systems* by Leslie F. Musk, Cambridge University Press, (1988), pg. 90. A satellite photograph (Nimbus-3) of an occluding front approaching Britian, July 25, 1969. The lines with symbols denote the warm front (triangles) and cold front (semi-circles). The lines without symbols show the coastlines of Ireland, Scotland, Wales and Normandy. The letter "L" denotes the minimum of pressure.

7.26 Scan-and-Decorate

A common scientific use of a drawing or painting program is to add modifications to an image that cannot be made in the original source. Sometimes, the source is Matlab or other scientific software, and the difficulty is that Matlab lacks the graphical capability we want.

It is equally common for the source to be a printed image which is input to the computer by a scanner. Since this can't be modified at all in its original form except by drawing, this is obviously an important application of drawing programs.

The figure below shows two versions of an illustration by a small child. The black-and-white draft was drawn in pencil. The colored version was made by scanning the pencil drawing and then modifying the result in KidPix Studio, which has a one-step command ("paint bucket") to paint the interior of a region defined by a closed curve with a given color. The bucket, like the paintbrush, can be modified to lay down a texture or pattern if desired.

Scan-and-decorate is literally child's play.

Figure 7.42: The lower half is a scan of a hand-drawing by Ian M. Boyd. The upper half is the scanned image after Ian imported it into Kid Pix and then colored it. He was 7.5 years old at the time.

Adobe provides several features that are helpful with scanned images. First, Illustrator can import EPS (Encapsulated Postscript) and Macintosh PICT images (as well as a number of other formats). These can be kept as part of the Illustrator drawing, or traced by hand. It is usually helpful to keep the imported image on a different layer from the non-imported objects. Once an image has been traced, it can be deleted by deleting the layer containing it.

Illustrator provides two features that are specifically designed to assist tracing. First, EPS images can be dimmed, that is, displayed in a page gray, making it easier to distinguish the trace from the imported image. Second, Illustrator has an Autotrace tool, which will copy a PICT image. The Autotrace tool is rather limited because complicated patterns of connected lines are often rendered as as lines that don't quite touch, forcing a lot of manual touch-up work.

Adobe therefore sells a stand-alone autotracing program called Streamline. This does a much more sophisticated job of tracing.

Unfortunately, we are still early in the age of machine intelligence. Consequently, a highly accurate, good-looking copy requires some manual work even with Streamline.

Scanned images are most useful when careful, time-consuming autotracing is not needed, such as when one of the following conditions is true:

1. The imported image does not require editing, but can be used, either by itself or placed in a more complex figure, as is.

2. Only a small part of the scanned image is needed.

7.27 Painting Tools

Painting programs and image-processing software adds some additional tools that are not usually found in drawing programs, or found only in a different form. For example, the paintbrush tool in a raster graphics program does not lay down a stroke and a fill, but rather paints a swath of bits with color. In Photoshop, the brush has two options that are not available in its Illustrator counterpart. First the brush can be "feathered", which gives soft, blurry edges. Second, the "opacity" of the brush can be varied. An opacity of 100%, the only choice in Illustrator, means the brush color is completely opaque and totally covers underlying features. Lower opacity allows previously painted features to show through the brushstroke. This is very valuable in imitating oil painting and watercolors where complex effects can be created by superimposing multiple layers of semi-transparent colors. It probably isn't too useful in most scientific applications.

The paintbucket tool exists in Illustrator, but only as a quick way of changing the fill of previously filled objects. The paintbucket tool in a painting program employs the same icon, but it really functions like pouring a bucket of paint on the figure: the color expands from the point where the cursor is clicked until the spread of color is blocked by lines or other objects drawn in a different color. Thus, if one is making a map and has drawn the complex coastline of Lake Michigan in black using the pencil, one can fill the lake with blue merely by making blue the active color, clicking the paintbucket icon on the toolbox icon, and clicking again in middle of the lake.

Painting programs offer other tools that have no counterparts in a drawing program. The Photoshop "smudge" tool is denoted an icon in the shape of a finger pressing on the paper. When the cursor is dragged on the image with this tool selected, the colors of a small area around the cursor are blended together — it is a sort of local Diffusion Filter. Ironically, a device for blending shades on a pencil or charcoal drawing is part of the standard toolkit for a conventional artist. Usually, the "smudger" is called a "stomp" and looks like a pencil except that the tip is actually made of tightly rolled paper.

It is easy to blend shades in pencil drawing, but much harder in painting where the colors may smear instead of blending smoothly. The "electronic stomp" exists only in painting programs, however, because color blending is a concept that is meaningful only in a pixel-based representation.

The "dodge" and "burn" tools allow one to lighten or darken small areas of an image. The "sponge" tool is useful only for color images where it increases or decreases the saturation of colors in a local region.

Collectively, the additional features of painting programs are useful mostly for creating complicated *textures*. As noted earlier, texture is only occasionally useful in engineering because the SHAPES are the crucial information. Most of the time, texture is only "chartjunk".

7.28 Rasterizing Objects or Photoshop-in-Illustrator

Objects in Illustrator can RASTERIZED, that is, converted into bitmap objects. Rather confusingly, there are four options: RGB, CMYK, grayscale and bitmap where the fourth option is "bitmap" in the narrow sense of 1-bit images (every pixel black or white). Illustrator allows one to specify several different resolutions.

The command Object > Rasterize (with an object selected) will pop up a dialog box. The "anti-alias" option makes for a smoother image, but also one that requires much more storage. An anti-aliased image may sometimes cause printer problems unless one's printer has a ton of memory.

The "create mask" option is very important unless the object is a rectangle. If the box is unchecked, then the rasterized image will be a rectangle circumscribing the object. When filters are applied, the filtered image will extend beyond the boundaries of the original object, perhaps all the way to the walls of the circumscribed rectangle. If "create mask" is checked, then only the object itself will be rasterized.

Why rasterize? The answer is that one can then apply a variety of filters that are meaningful only for bitmap images. Illustrator itself comes with only a dozen or so image filters, grouped as three "GE (Gallery Effects)" submenus under the Filter menu. However, all the filters of Adobe Photoshop can be applied, provided aliases (or copies) of these Photoshop filters are stored in the Plug-Ins folder of the Illustrator folder.

To an engineer, Diffusion and Anti-Diffusion filters (not available in Illustrator's own sets) are the easiest to explain. Given a discrete array of colors (or temperature values or whatever), diffusion can be applied by adding the a constant times the AVERAGE of the four nearest neighbor values to each pixel. This is a simple centered finite difference approximation to the diffusion equation. Sharp, small features are smoothed.

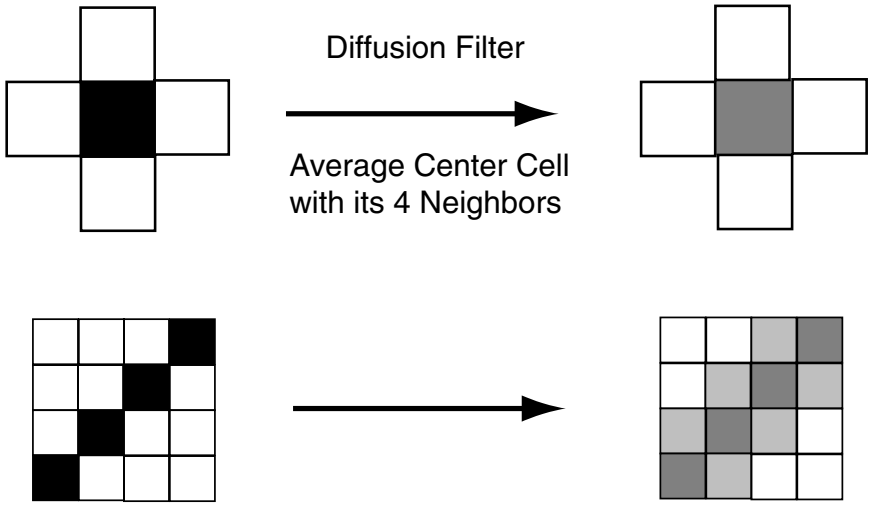


Figure 7.43: Schematic of a diffusion filter. It is applied by taking some weighted average of a cell with its four nearest neighbors as shown in the top half of the figure. The bottom shows what happens when this filter is applied to a rasterized figure with a line: the line is blurred.

An Anti-diffusion filter is also popular in image processing. By subtracting the average, i. e., the opposite sign from diffusion, one can enhance small scale features. Edges are enhanced and amplified, making it easy to recognize the edges of features that were blurry in the original image. When the raw pictures were obtained from an

astronomical telescope or CAT scan, blurred by instrumental noise which rather acts like a diffusion, the Anti-Diffusion undoes some of the instrumental diffusion to yield a sharper, clearer filtered image.

The Diffusion and Anti-Diffusion filters can only be applied after rasterization because diffusion has no meaning for a vector, i. e., a line segment or curve, and therefore is meaningless in vector-based graphics. The filter only makes sense when applied to a block of pixels which are a discrete approximation to a continuously varying image.

Fig. 7.36 shows how the Diffusion filter is applied. Sharp, black lines are blurred into wider rivers of gray by diffusion. For this reason, the Diffusion filter is named "Blur" (as a tool) and "Unsharpen" (as a filter) in Adobe Photoshop.

Fig. 7.37 shows a potpourri of other filters, applied to a common object. There is nothing special about these particular filters. The point is that filtering of a rasterized object can help to rectify one of the limits of drawing: the lack of a natural-appearing texture on objects.

In engineering drawing, usually texture is undesirable; it distracts from the shapes and graphs that carry the cognitive message. However, diagrams in *Scientific American*, *National Geographic* and some undergraduate textbooks have illustrated the power of embedding complicated ideas in a professionally-drawn (or professionally-painted) detailed background to provide a context. It is easier to understand the layout of the Lost City of the Mayas (or whatever) from a painting in aerial perspective than from a wireframe diagram.

Such realistic images are usually drawn by a professional artist. However, the artist begins with a sketch drawn by an engineer or scientist. So, it is important that the technologist be able to draw at least well enough so that the artist can understand the sketch.

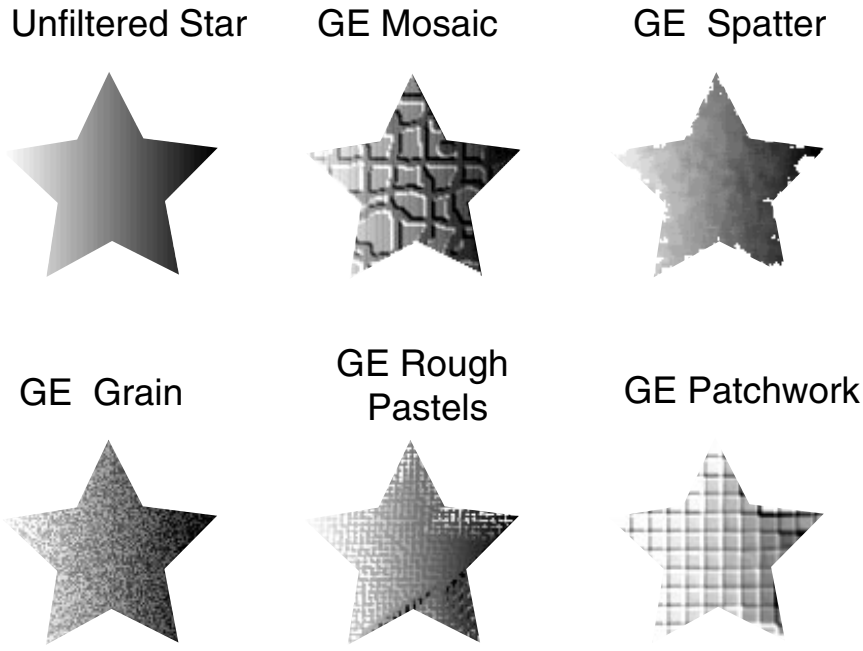


Figure 7.44: A sampler of the effects of various Adobe Illustrator GE ("Gallery Effect") filters on a rasterized star with a grayscale gradient.

7.29 Drawing Theory and Drawing Programs

A major theme of how-to-draw books is that complex shapes are built up from simple shapes such as circles, cones, squares and rectangles — precisely the kind of shapes that drawing programs are good at generating. Fig. 7.42 shows a third-grader's drawing of a horse. Although crude and unfinished, one can see how the boy has used two large circles to organize the torso of the horse.

Fig. 7.43 illustrates the theory of building up drawings from complex shapes. The first step is to draw assemblies of simple shapes to organize the drawing. In addition, guidelines (dashed) are also drawn. At later steps, the details are added. Many parts of the original squares, circles, and so on and all of the guidelines must be erased or overwritten to create the final, finished shape.

Erasing guidelines is both a time-consuming nuisance and a technical challenge in pencil or ink drawing. However, Illustrator and other drawing programs make it easy.

First, preliminary shapes can be placed on a separate layer from the real figure. The guidelines can be drawn in a pale gray or a different color than used in the figure so that the guide-shapes can be easily distinguished from the real thing. When one is finished, one can lock the layer, and then delete it at the end.

Second, Illustrator allows one to convert graphical objects to what Illustrator calls "guides". These are non-printing figures that appear as fixed, immoveable patterns of dotted lines on the screen. An architect who wants to draw a three-side octagonal bay on the front of a house could first draw an octagon of the correct size and place using the Polygon tool, then convert the octagon to a guide. One can draw the three segments of the bay on top of the guide. The guide does not need to be erased, but there is less flexibility in altering it or coloring it than when the guides are ordinary objects on a different layer.

In any event, the shape tools of a drawing program simplify the creation of guides; one has a couple of different options for exploiting them.

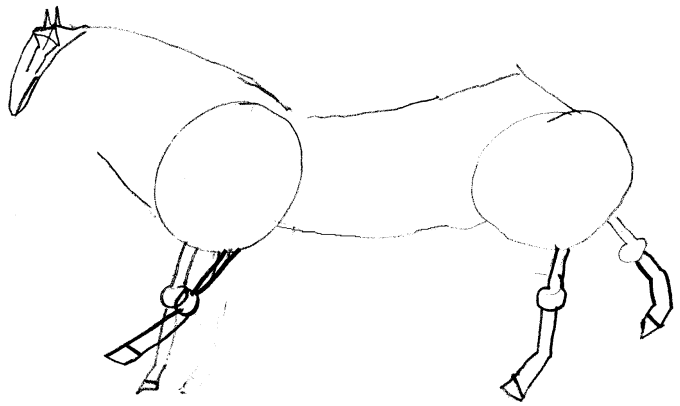
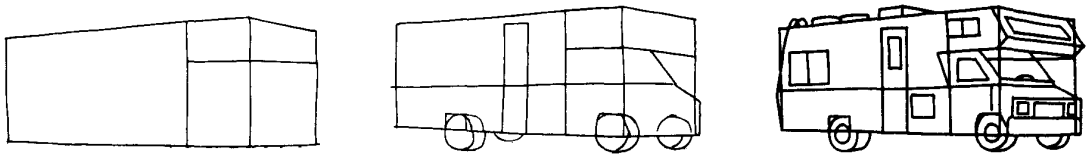
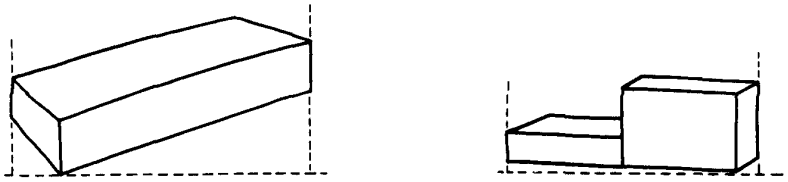


Figure 7.45: Pencil drawing of a horse by Ian M. Boyd, who was eight years old. Even a child can understand the theory of building complex shapes from simple shapes; note the two large circles in the body. If the drawing had been finished, these circles would have been erased.

1 ***Draw the first few steps lightly in pencil, and only erase after you have made all of your corrections and additions.***



2 ***The bodies of all of the vehicles on these pages are made of three-dimensional shapes. In order to draw them correctly, you'll need to give yourself guidelines such as the ones below. This can be done without a ruler.***



3 ***Be sure to draw guidelines for the wheels and tracks in each drawing.***

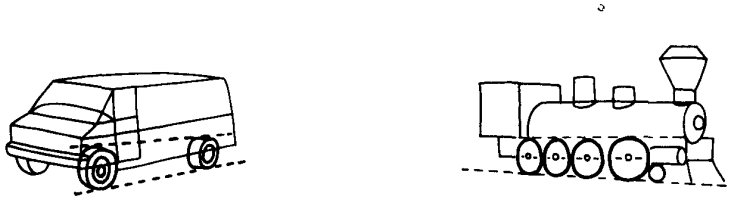


Figure 7.46: Strategy in a how-to-draw book. The rest of the volume consists of several stages in each drawing. The early stages are patterns of rectangles and circles in perspective; much of these organizing shapes are erased or overwritten by details in finishing the drawing. Scanned from *I Can Draw Cars, Trucks, Trains and Other Wheels* by Tony Tallarico, Simon and Schuster, (1981), pg. 2.

Computer object-oriented graphics programs are also good at implementing some other crucial ideas of drawing theory. For example, to maintain consistency in shading, most monochrome (single color) drawing is done using a "four-value" system (Fig. 7.44). The pencil artist obtains black and two shades of gray by switching pencils, varying pressure, and tilting the pencil on its side. The ink draftsman must fake shades of gray through patterns of cross-hatching. The computer artist can obtain perfect, consistent shades of gray merely by choosing gray swatches in the Paint Style palette.

Drawing programs almost invariably have an "Eyedropper" tool. When touched to a region, it alters the Paint Style palette to match the region. Thus, it is easy to consistently apply subtle shades of color in computer art. In four-value monochrome drawing, though, one hardly needs the Eyedropper.

"Contrast" is the difference between the light and dark objects in a picture; it is the black-to-white range. In a high contrast picture, the white is really white and black is black. In low contrast, the white is a pale grey and the black is a very dark gray. It is easy in computer drawing to experiment with contrast. Illustrator has a Select-by-Color command which can be used to change any of the four grayscale values.

The "key" of a picture is the median of the grayscale values. A "high key" figure is one whose overall values are rather pale, or in other words one in which both intermediate grays are light. A "low key" figure is predominantly dark. This, too, is easily changed.

Raster graphics program such as Photoshop have filters specifically to alter "contrast" and "key". Thus, in both object-oriented and bitmap graphics, one can change these properties easily.

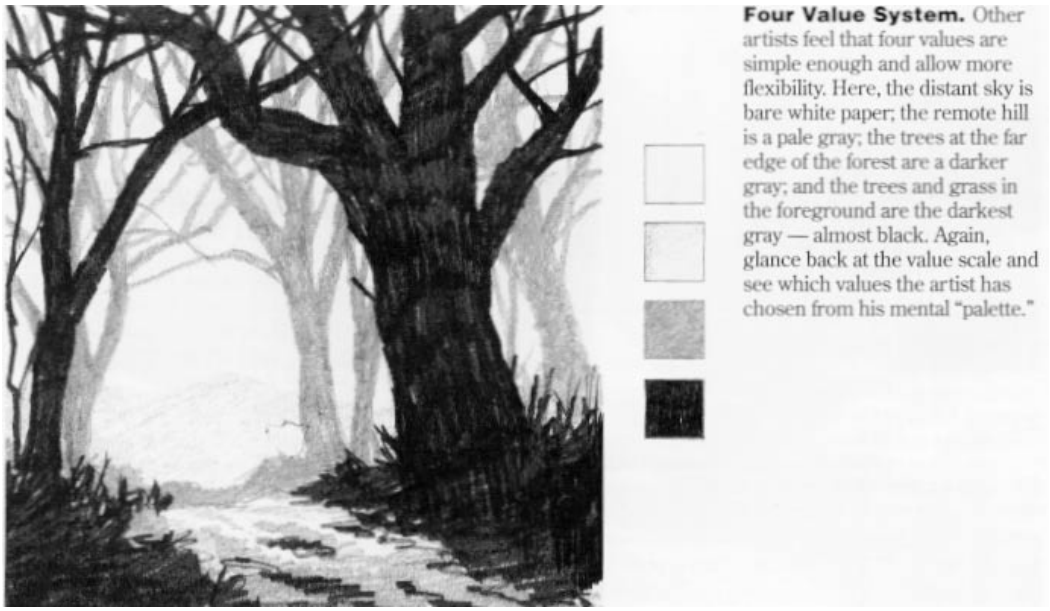


Figure 7.47: Although pencil and charcoal may in theory create an almost infinite ranges of grayscales from black to white, in practice most artists use only four grayscales — white, black, and two shades of gray — to create a monochrome drawing. This figure illustrates the "four-value" system. As noted by Blake and Petrie, many artists go further by employing a "three-value" scheme with only a single shade of gray. Scanned from *Getting Started in Drawing* by Wendon Blake with drawings by Ferdinand Petrie, Northlight Books, Cincinnati (1991), pg. 24

THE ROLE OF CONTRAST AND KEY

High Contrast. When you study your subject and choose your values from the scale, it's important to decide how much contrast you see between the values of nature. On a sunny day, you'll see a high-contrast picture. That is, you'll see bright whites, dark grays or blacks, plus various shades of gray between the two extremes. Another way to put it is that the subject gives you a *full range of values.*

Low Contrast. On an overcast day, the same subject will present a much more *limited value range.* The light areas of the picture aren't as bright and the darks aren't as dark. The whole picture consists of grays: a pale gray for the lightest area, a not-too-dark gray for the darkest areas, and a couple of medium grays in between. Note that the values of a high-contrast picture are widely scattered on the value scale, while the values of a low-contrast picture are close together on the scale.

High Key. Artists also talk about the *key* of the landscape. When the overall value of the landscape is generally pale, it's called a *high-key* picture. In this case, the values are apt to come from the beginning of the value scale — pale grays and possibly white. Even the darkest areas in the picture — the trees on the shoreline at the left — are relatively pale. Foggy coastlines and misty landscapes are often in a high key.

Low Key. A *low-key* picture is predominantly dark, drawing most of its values from the dark end of the scale. A change in light or weather can convert the same subject from a high key to a low one, as you see here. Now the trees are a deep gray (almost black) and the lighter areas of the sky, water, and shore are still fairly dark. Only the strips of light in the sky and water come from the light end of the value scale. Note that the high-key picture at left is a low-contrast subject, while this low-key landscape has higher contrast.

Figure 7.48: Scanned from *Getting Started in Drawing* by Wendon Blake with drawings by Ferdinand Petrie, Northlight Books, Cincinnati (1991), pg. 24

7.30 Logos and Overheads: Cosmology According to Michael Turner

"Nelson Algren, the Chicago writer, said he lived by just three rules, two of which could be listed in polite company: Never play cards with any man named 'Doc' and never eat at any place called "Mom's.' Viewers of the face-of called 'Great Debate III: Cosmology Solved?' — pitting the University of Chicago's Michael Turner against Princeton University's James Peebles in a crowded auditorium on 4 October — might have come up with another law, to be violated at great peril: Never debate a cosmologist whose viewgraphs are considered objets d'art."

— James Glanz, *Science, 282*, 1247-1248 "Does science know the vital statistics of the cosmos?" (1998).

Fig. 7.46 illustrates the logo advertising a debate between two cosmologists in honor a third cosmologist, David Schramm, who was killed while flying a light plane a few months earlier. Turner is a legend for his flamboyant transparencies; two samples are shown in Fig. 7.47. His adversary, James Peebles, is not flamboyant at all. However, he almost won the Nobel Prize for work done while still in graduate school when he and his advisor, the late Robert Dicke, explained the observations that did win the Nobel Prize for Penzias and Wilson. He also wrote the book on cosmology (literally!). Ability counts, too, which is something to keep in mind while in the throes of graphomania.

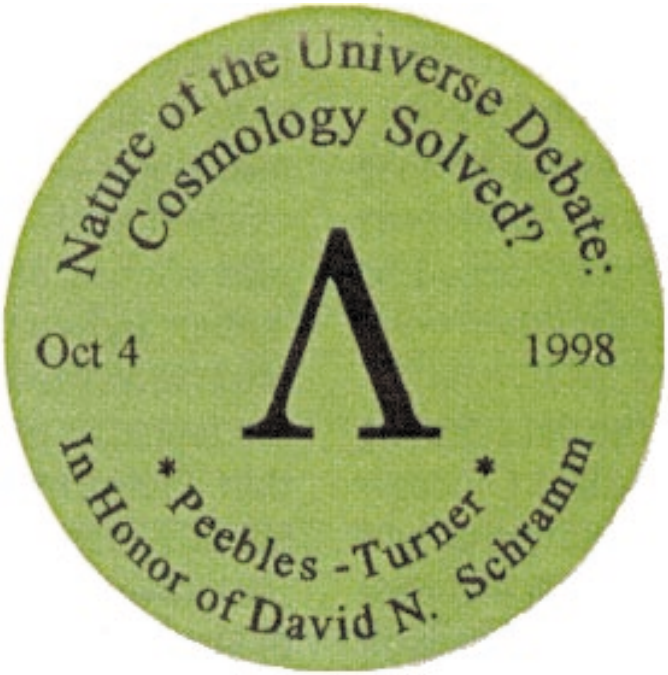


Figure 7.49: Logo for the announcement of a debate between Michael Turner and James Peebles in 1998. The Greek letter Λ is the symbol for the "cosmological constant"; Turner argued that it resolves the major problems in cosmology. A good application of a "path-area" tool to wrap text around the inside of the bounding circle.

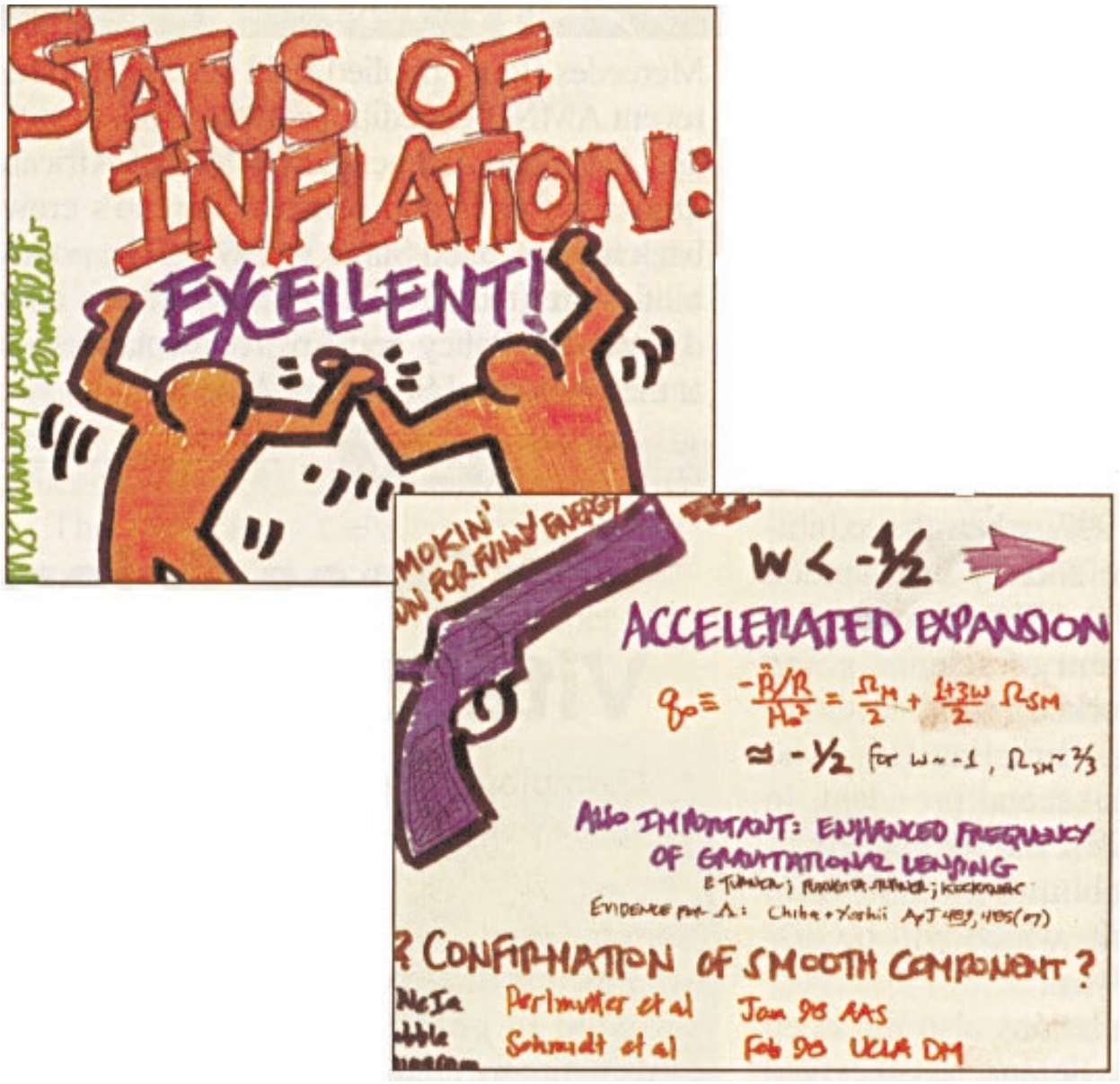


Figure 7.50: Two of Turner's flamboyant transparencies for his debate with James Peebles.

7.31 Summary of Illustration Program Ideas

Generic ingredients of a good drawing program

- Multiple object-selection tools: for the entire curve, for an individual anchor point, for a previously-grouped collection of elements.
- A group procedure to link elements into a single entity that can be moved, resized and rotated as if it were a single element.
- Multiple layers, each of which can be individually locked or unlocked, shown or hidden, traced and then erased.
- Zoom and grab tools for shifting the view of the image while it is being drawn.
- Line-drawing curves:
  (a) straight line segment tool
  (b) curve-drawing tool with Bezier interpolation
- Paintbrush to lay down stripes of color and texture.
- Diverse fill capabilities: solid colors, gradients of color or grayscale, pre-stored texture patterns, and patterns created by the artist.
- A join operation to connect two curves into a single curve.
- Scissors and knife operations to cut line segments or polygon sides and cut cracks into polygons or other objects.

7.32 Features That Are Useful in Scientific and Engineering Graphics

- Adding different patterns of shading (light stippling, dark stippling, diagonal lines, etc.)
- Arrows, added to axis lines or used to connect text boxes with data curves.
- Attaching text to a curving path to make it fit a crowded space and also to make it clear what element is labelled by the text.
- Inserting text into an irregularly shaped box to make it fit easily.
- Making wavy lines (Zigzag filter in Illustrator).