Freeform Surface Modeling

There are two methods for creating freeform surfaces in HP PE/SolidDesigner: blending and lofting. This article describes the basics of lofting. The geometry engine, which implements the lofting functionality, uses a single-data-type implementation for its geometric interface, but takes a multiple-data-type, hybrid approach internally.

by Michael Metzger and Sabine Eismann

HP PE/SolidDesigner's kernel functionality consists of several modules that communicate through well-defined interfaces, supported by logical class definitions and hierarchies. In Fig. 1, for example, the geometric data interface for the topology engine (the Boolean engine, see article, page 74) consists of three basic elements (points, curves, and surfaces) and the corresponding utility functions like intersections. This technique makes it easy to add new functionality. For example, introducing new geometry data types is just a matter of delivering all member functions of the geometric interface for the new geometry type.

The implementation of such a concept looks simple, but reality has shown that it takes a lot of effort to keep the interface clean and to avoid copying and converting data. This is especially true for data having connections on both sides of the interface, such as pieces of a curve or curves on a surface.

The Geometry Engine

In designing a completely new implementation of the geometric kernel for a solid modeler one has a chance to avoid the problems of older implementations. What are the real problems of existing implementations? There are two fundamental approaches: NURBS libraries and hybrid methods.

NURBS libraries have only one data type: NURBS, or nonuniform rational B-splines. This data type can represent all analytics (like planes, cylinders, spheres, etc.) exactly. This means that complex freeform surfaces as well as simple analytics are represented with one single data structure. The geometrical problems only have to be solved for this single type. This sounds promising, but it turns out that the algorithmic stability does not satisfy the requirements of HP

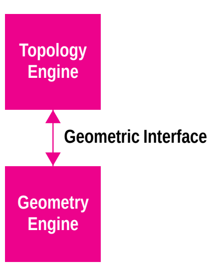


Fig. 1. The HP PE/SolidDesigner topology and geometry engines communicate through a well-defined geometric interface.

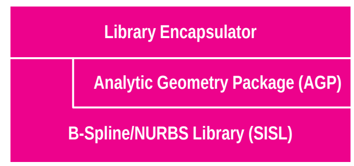


Fig. 2. HP PE/SolidDesigner geometry engine.

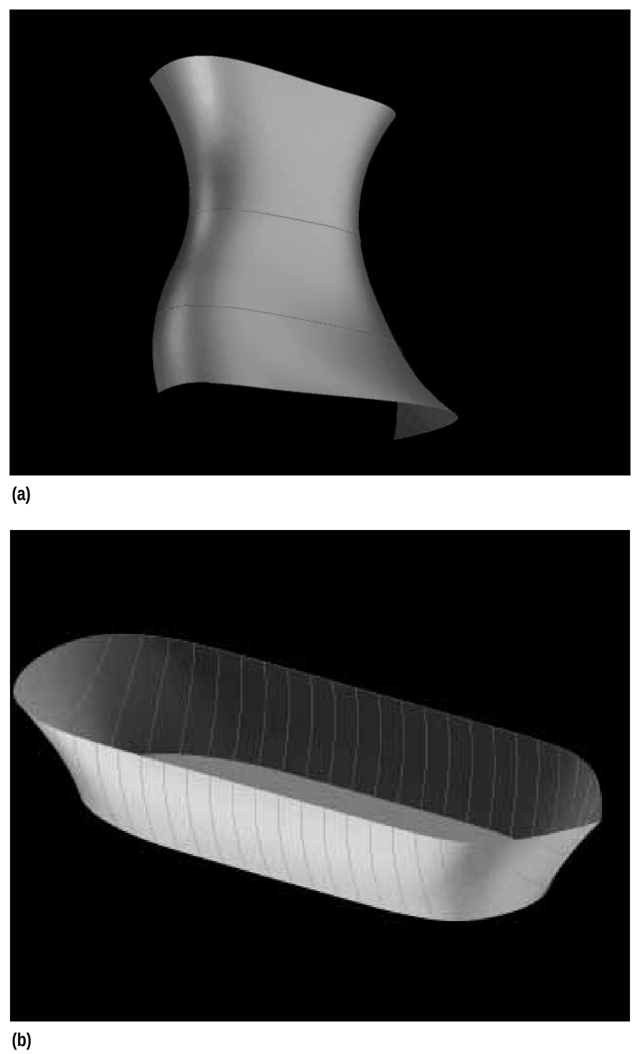
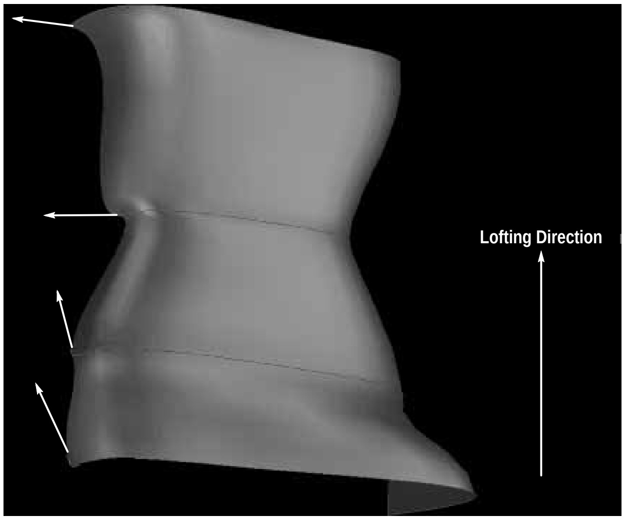
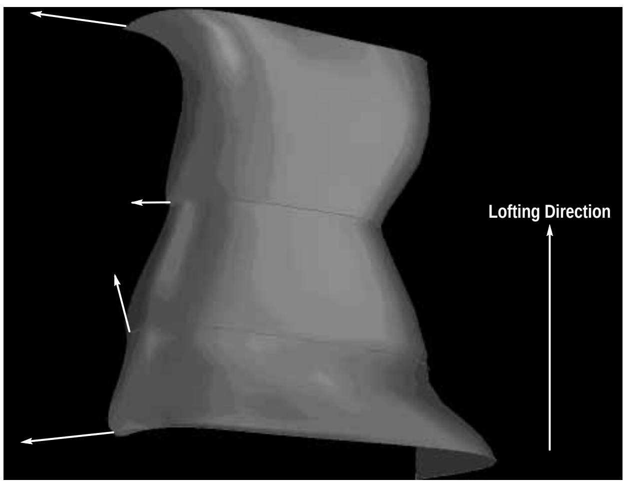


Fig. 3. (a) A lofted surface. (b) Lofting originated in ship design.



(a)



(b)

Fig. 4. Specifying the tangent profile, a kind of vector field along a curve, influences the shape of a surface.

PE/SolidDesigner. In addition, the performance is poor, especially when analytic surfaces (represented as NURBS) are intersected.

Hybrid methods are used in the HP PE/ME30 kernel (Romulus). All possible geometry data types are available, and clever special case handling results in high performance. The disadvantage is that the introduction of a new data type is an enormous effort. In addition, the Romulus kernel doesn't distinguish cleanly enough between geometry and topology, so building new functionality on this kernel can be very cumbersome and error-prone.

In HP PE/SolidDesigner we tried to combine the advantages of both approaches. The advantage of a NURBS library (one data type) is realized in the class hierarchy of HP PE/SolidDesigner: the geometric interface knows only points, curves, and surfaces. For the internal geometry structure the hybrid method was chosen. Data types include analytic types (plane, sphere, cylinder, cone, torus), semianalytic types (parallel swept B-spline, spun B-spline), B-splines, and NURBS as an extension of B-splines.

Fig. 5. Multiply connected curves.

As shown in Fig. 2, HP PE/SolidDesigner's geometry engine consists of three parts: the library encapsulator, the analytic geometry package (AGP), and the B-spline/NURBS library (SISL).

The library encapsulator delivers many convenience functions for the geometric interface and ensures its integrity. All functions dealing with geometry have to pass through the geometric interface. The only exception is a small part of the blending algorithm, which for performance reasons bypasses the library encapsulator and calls SISL directly.

The AGP was developed by DCUBED Ltd. of Cambridge, England and SISL was developed by the Senter for Industrieforskning of Oslo, Norway.

Freeform Surface Modeling

There are two methods for creating freeform surfaces in HP PE/SolidDesigner: blending and lofting. The remainder of this article describes the basics of lofting.

Lofting means the (exact) interpolation of a set of points or curves by a smooth curve or surface. Fig. 3 shows examples of lofting. Lofting originated in ship design and was used a long time before computers were invented.

The mathematical solution of this problem leads to the definition of splines. There are many spline types, each having its specific advantages and disadvantages. The most common spline types are Bézier splines, B-splines, and NURBS.

For CAD applications the most general splines are NURBS, since they can represent analytics exactly. This can be important when it comes to intersections of splines and analytic surfaces. B-splines are NURBS with all weights equal to 1. They are more stable and faster in intersections but cannot represent analytics (except the plane) exactly. B-splines are made up of a sequence of Bézier pieces, connected according to their continuity at the transition points. We won't go into detail concerning spline mathematics here since there is abundant literature on this topic.[1,2]

In addition to the pure interpolation of points and curves, lofting allows the definition of tangent profiles at each 3D

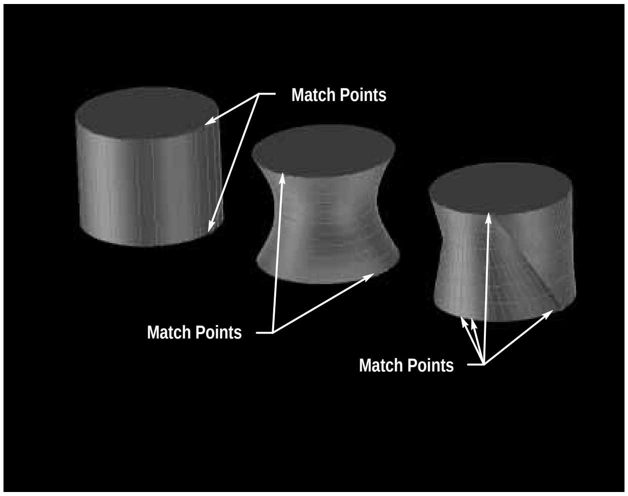


Fig. 6. Reparameterizing surfaces by specifying matching points on different input curves.

curve. A tangent profile is a kind of vector field along the given curve, as shown in Fig. 4. Both the directions and the lengths of the tangents influence the shape of the surface.

In practical applications the user normally wants to interpolate not only a series of single curves but also a series of multiply connected curves, as shown in Fig. 5. For this purpose HP PE/SolidDesigner connects the incoming profiles to a single B-spline curve. It is not required that a profile be smooth; it only needs to be C^0 continuous (closed). The C^0 locations in the profiles later correspond to edges in the complete model.

In addition to tangent profiles, the parameterization of the input curves is another important factor determining the shape of the lofted surface. In HP PE/SolidDesigner, parameterization can be influenced by splitting the input curves at arbitrary points (match points) and defining different length ratios in the subsequent profiles (Fig. 6). Within a curve segment, HP PE/SolidDesigner tries to create a parameterization according to the chord length of the curve (chordal parameterization).

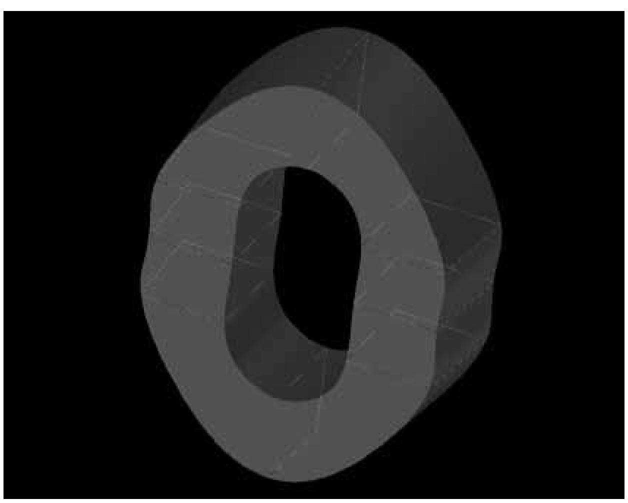


Fig. 7. Closed (periodic) surface created using lofting.

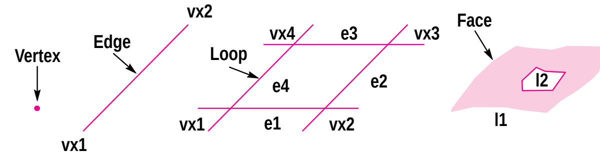


Fig. 8. Topological elements: vertex, edge, loop, face.

It is also possible to create closed (periodic) surfaces using lofting. In this case the first and last profiles are identical (Fig. 7).

Topology

Before explaining how topology is attached to the loft geometry, some definitions are needed (see Fig. 8):

- A vertex lies on a 3D point and can be viewed as the corner of a face.
- An edge is a bounded portion of a space curve. The bounds are given by two vertices.
- A loop represents a connected portion of the boundary of a face and consists of a sequence of edges.
- A face is a bounded portion of a geometric surface in space. The boundary is represented by one or more loops of edges.

Given a B-spline surface obtained from the spline library using the profile interpolation method, topology has to be built on this surface to get a loft body. As a boundary for the face, a loop consisting of four edges is created (Fig. 9). The edges lie on the first and last interpolation curves (e1 and e3) and on the left and right boundaries of the B-spline surface (e2 and e4).

The interpolation profiles don't have to consist of only one curve per profile. For more complex shapes different curves can be combined in a profile. It is necessary to generate a face for each matching set of curves. One way to do this is to use lofting to create a B-spline surface for each matching set and then build the appropriate faces on these surfaces. Because there is no exact specification of how the left and right boundaries of these B-spline surfaces should look there may be gaps between the faces (Fig. 10a). This would lead

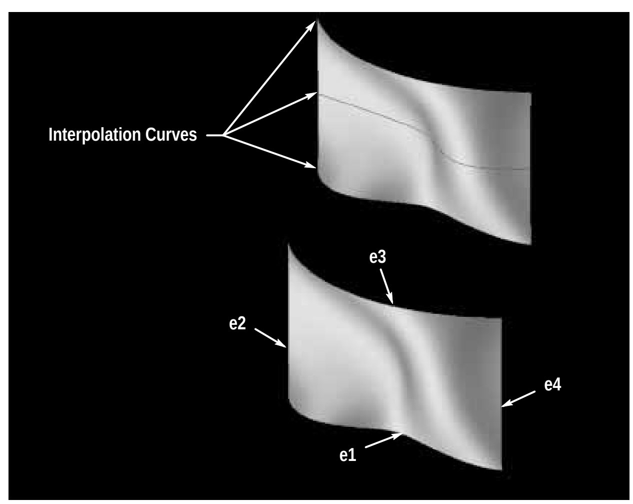


Fig. 9. Building topology on a B-spline surface by creating a boundary consisting of four edges.

to an illegal body, since all neighboring faces in a body have to share a common edge. When there are gaps between the faces no common edge can be found and it isn't possible to generate a valid body.

To eliminate gaps, the curves in one profile are joined temporarily and only one loft surface is generated. This B-spline surface then is split into appropriate parts at the start and end points of the interpolation curves. The faces are then built on the split surfaces. This ensures that there is no gap between the faces.

To match the correct curves or the correct portions of the curves it is necessary that all curves in a matching set have the same parameter interval. This is ensured by reparameterizing all curves belonging to the same matching set to the same (arbitrary) parameter interval. After this all curves of a profile are joined and the joined curves then automatically have the same parameter interval.

A valid solid body must describe a closed volume. For this reason only closed interpolation profiles are used. From these the lofting facility will generate faces forming a tube, which still has two open ends (Fig. 11). For each of the two ends a planar face is added. Theoretically these top and bottom faces can lie on any type of surface as long as the first and last interpolation profiles lie on the respective surfaces.

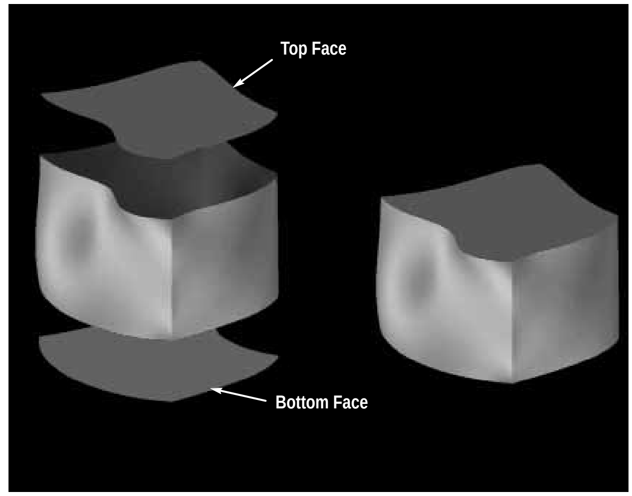


Fig. 11. Lofting generates a tube. Endfaces are added to make a solid body.

Lofting in HP PE/SolidDesigner

The spline library allows arbitrary 3D curves in space as interpolation profiles for lofting. To simplify the input process for the user, only planar profiles are allowed in the current release. These planar profiles can easily be generated in a workplane using 2D creation methods. All workplanes containing the profiles are gathered in a *workplane set*. The user specifies which set of curves should match in lofting. Different matching specifications will produce different loft results (Fig. 12).

Because the spline library only accepts B-spline curves as interpolation curves the analytic curves in the profiles have to be approximated by B-splines. Another reason for this is the above-mentioned joining of curves in a profile to obtain only one B-spline surface.

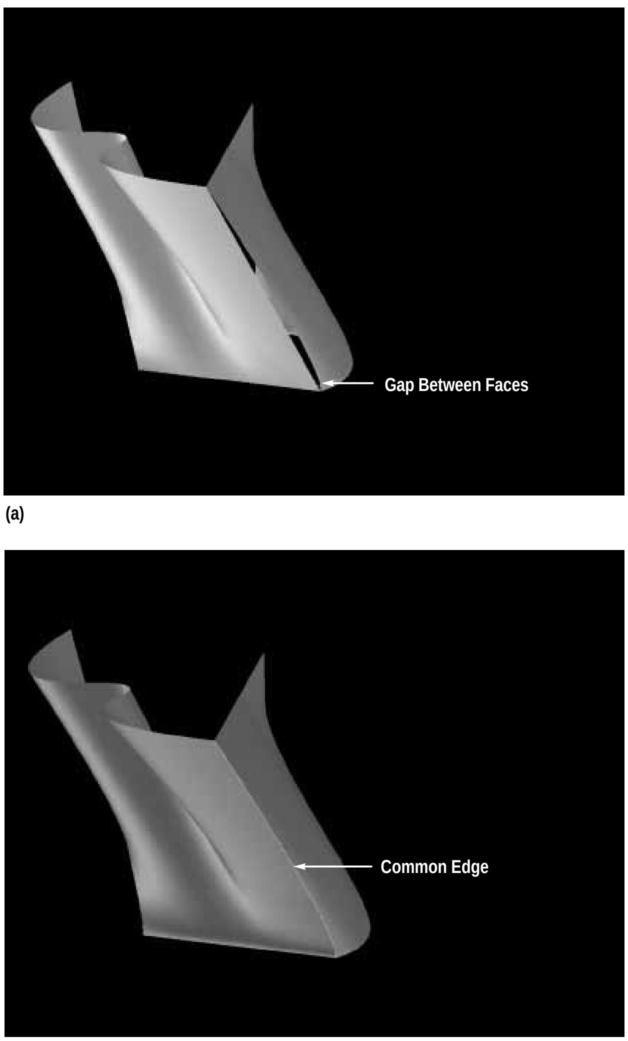


Fig. 10. (a) Illegal body with gaps between faces. (b) The system generates a common edge to eliminate gaps.

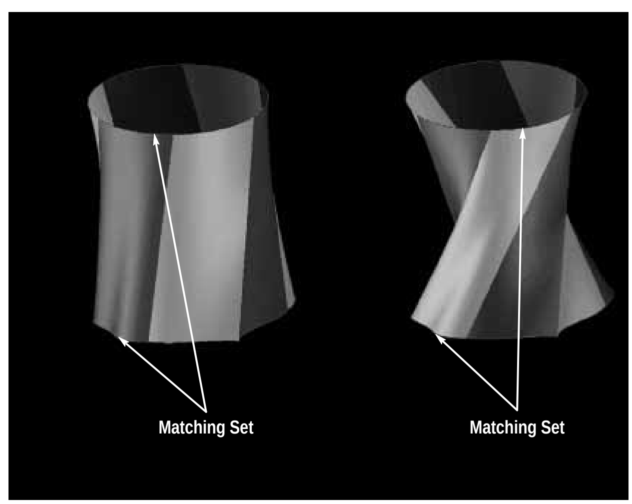


Fig. 12. Changing loft results by specifying different sets of matching curves.

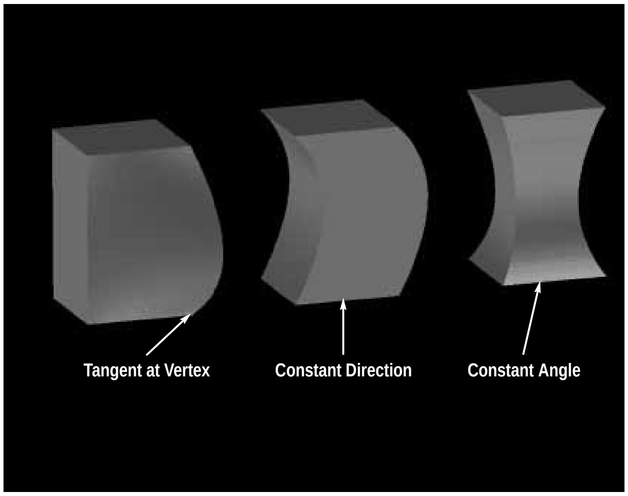


Fig. 13. Adding tangent conditions to change the shape of a body. (a) By defining a tangent direction to one or more vertices in the profile. (b) By specifying a constant direction for the entire profile. (c) By specifying an angle at one point of the profile. This angle is kept constant along the entire profile.

Another possibility for influencing the shape of the loft body is to add tangent conditions. In HP PE/SolidDesigner there are three different methods for doing this (Fig. 13):

- Define a tangent direction to one or more vertices in the profile.
- Specify a constant direction for the entire profile.
- Specify an angle at one point of the profile. This angle is kept constant along the entire profile.

For topology creation, especially face generation, the curves underneath the bounding edges of the faces have to be determined. Because the lofting algorithms only generate one B-spline surface, to get a properly connected tube this surface has to be split somehow. Because the single curves on the profiles have already been reparameterized to the same parameter interval for correct matching, this knowledge can be used to split the B-spline surface correctly. The boundaries of the split surface all lie on *isoparametric curves* of the loft surface. An isoparametric curve is a curve on a surface that has a constant u or v parameter value. In our case the loft direction is the v-parameter direction of the surface. This means that the left and right boundary curves of the faces are v-isoparametric curves. Splitting the surface along the

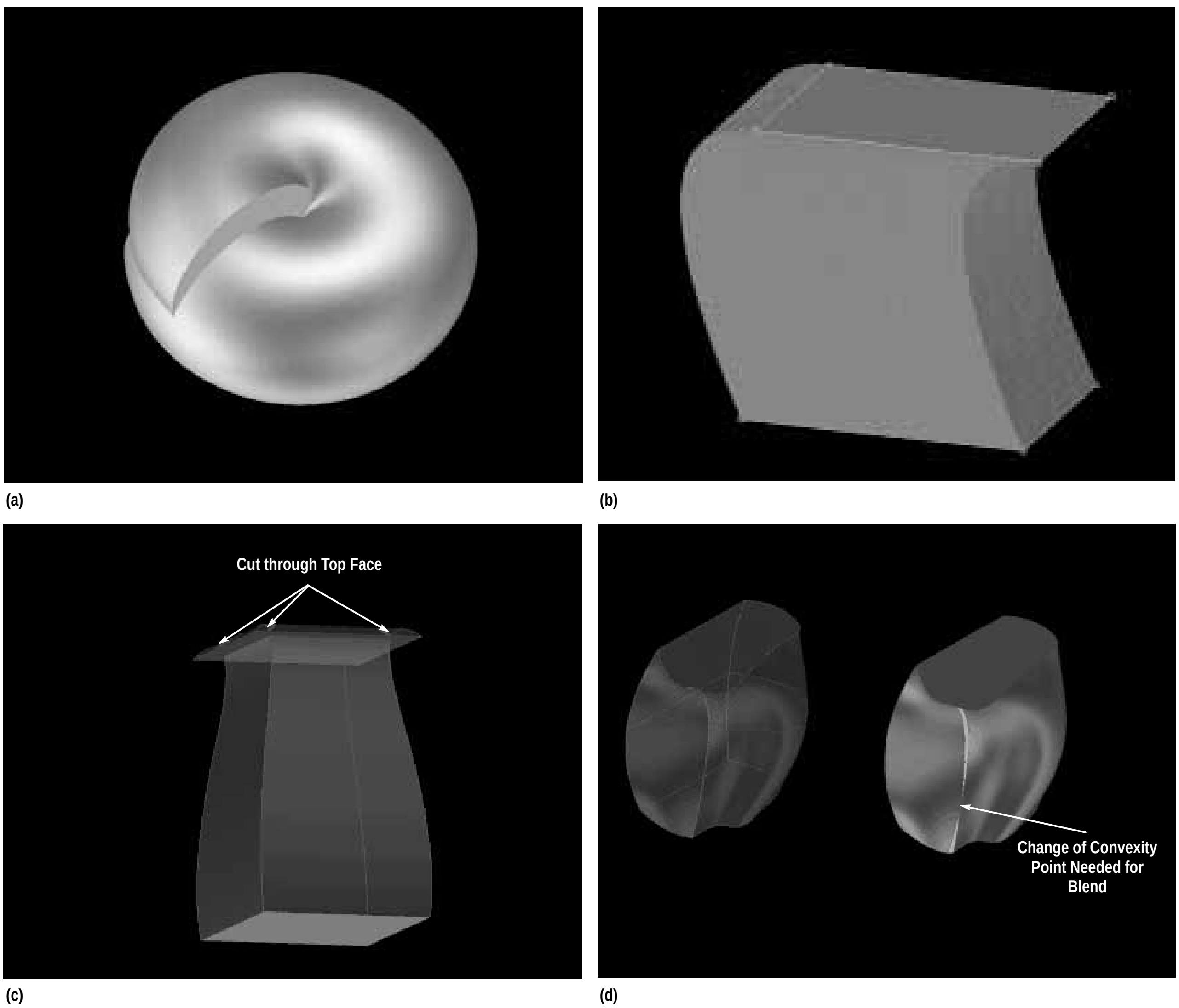


Fig. 14. HP PE/SolidDesigner checks for various properties that solid bodies shouldn't have. (a) Self-intersecting body. (b) Vanishing normals or derivatives. (c) Intersection with the top or bottom face. (d) Change of convexity at an edge.

v-parameter values of the start points and endpoints of the interpolation curves will result in the desired subsurfaces. The edges created on these v-isoparametric curves are always common to two neighboring faces.

Analytic Surface Type Detection

From a mathematical point of view the interpolation task that constitutes lofting is finished when the B-spline surface is created. From a CAD user's point of view the work is only partially finished. The reason is that it very often happens that a lofted body contains B-spline surfaces that represent analytical surfaces, mostly planes and cylinders. A CAD user wants to recognize these analytics in later processes for easier control in manufacturing. Data size, intersection performance, and stability are much better when dealing with analytics rather than approximated geometry. For these reasons a clever analytic detection algorithm is implemented in HP PE/SolidDesigner which replaces the B-spline strips by analytics after the B-spline creation and before the final topology is built.

The algorithm is based on the geometry of the input profiles. If curves of the same type are matched the basic definitions of these curves are compared (for example, the center of a circle, its radius, its starting point, etc.). Then, starting from the first two profiles, a corresponding analytic surface is built. In the next steps the other curves along the profiles (in the loft direction) are examined to see whether they fit this surface. If they do, the corresponding B-spline strip is exchanged and the neighboring topological information is adopted. This is done for each curve in the profile loop. Since the algorithm is based on the profiles and not on the lofted B-spline surface it is extremely fast and takes less than 1% of the time required for the lofting operation.

Special Cases

Lofting is a powerful tool for creating freeform surfaces in HP PE/SolidDesigner. On the other hand, there is a danger of creating surfaces that are not manufacturable or that have properties that can cause problems in later operations. For this reason, HP PE/SolidDesigner applies extra checks to ensure that the result of lofting is a clean body. These

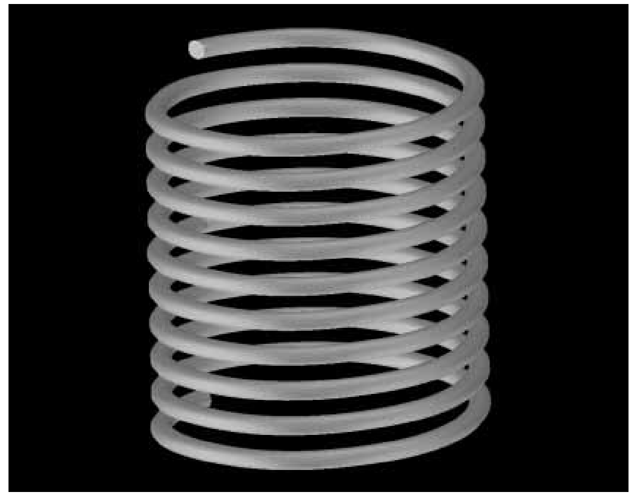


Fig. 16. Spring created using the workplane inclined command.

checks take extra time, normally more than the creation of the surface itself. HP PE/SolidDesigner therefore offers a button on the user interface to switch off these checks. It makes sense to switch the tests off in the surface design phase. For the final acceptance, however, it is recommended that the tests be run, since a corrupt model cannot be repaired later.

In the following examples we show the various properties a solid model shouldn't have. HP PE/SolidDesigner checks all of them and rejects the lofting operation if at least one of them appears. In the preview mode, the user can examine the object to find the root cause for the problem. The forbidden properties are:

- A self-intersecting body (Fig. 14a)
- Vanishing normals or derivatives (Fig. 14b)
- Intersection with the top or bottom face (Fig. 14c)
- Change of convexity at an edge (Fig. 14d). This test is always done and ensures that the specific edge can be blended later. HP PE/SolidDesigner will insert a topological vertex at the place where the convexity changes.

Practical Experience with Lofting

The most critical point in using lofting is the proper definition of the profiles and the workplanes. It turns out that in many real-life applications the profiles do not vary at all (e.g., helical constructions) or only a little. HP PE/SolidDesigner supports these surface classes by offering special

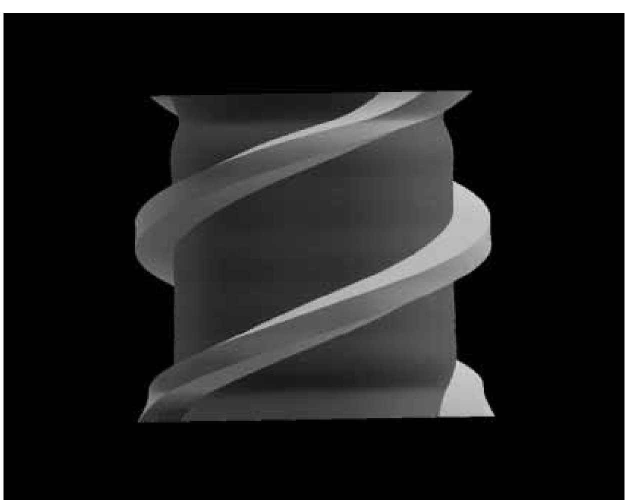


Fig. 15. Drill created using a special HP PE/SolidDesigner command to define a set of parallel workplanes, each turned by a given angle around an axis orthogonal to the base workplane.

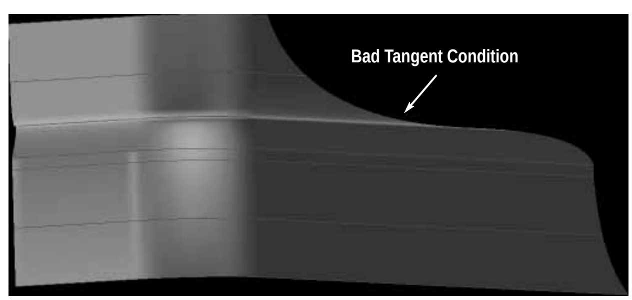
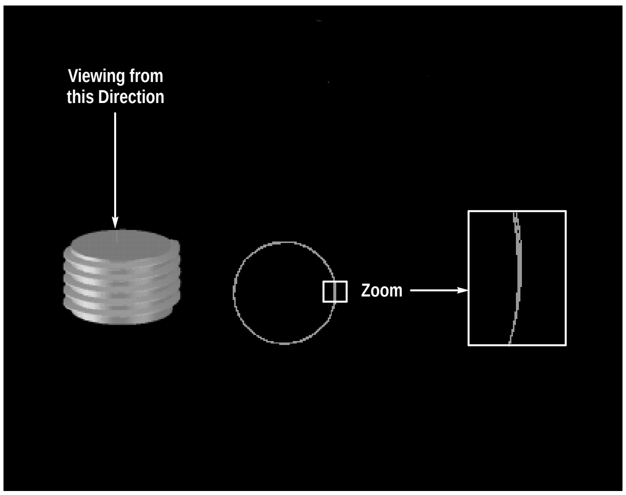


Fig. 17. Positioning too many profiles over too short a distance results in a wavy surface.

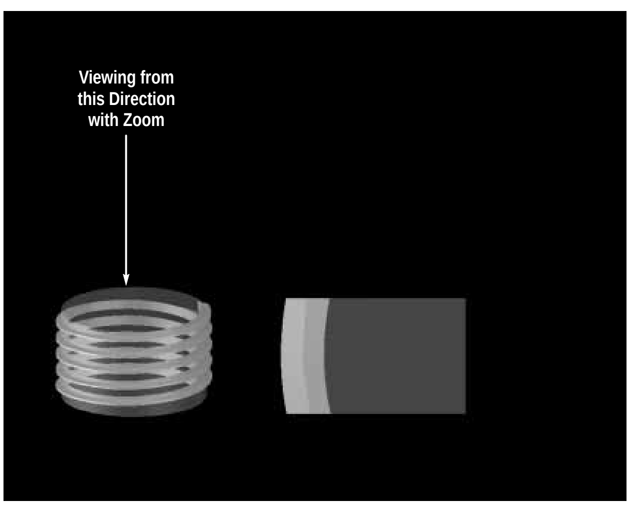
commands to create series of workplanes in the 3D space. These commands let the user define a set of parallel workplanes, each turned by a given angle around an axis orthogonal to the base workplane. Objects like drills can be created very easily (Fig. 15).

Using the "workplane inclined" command defines a set of workplanes at an angle to the base workplane. This is a way to create springs and other helical shapes (Fig. 16).

These special commands do not help in all situations. Sometimes the complete workplane set has to be defined by hand. Here it is important to know some basic behavior of the lofting algorithm to avoid subsequent problems with the Boolean topology engine. Often lofting is not used to create a completely new body but to cut off some existing geometry (loft remove) or to fill gaps (loft add). The most important property of lofting the user must keep in mind is that the surface starts oscillating if too many conditions (profiles, tangency conditions) are specified on too short a distance. The term "short" means short relative to the total object size. Positioning ten profiles over a distance of 100 millimeters causes no problems. Doing the same over a distance of one millimeter creates an awful surface. The same is true for the complexity of the profiles and the way the profiles change from one workplane set to another (Fig. 17).

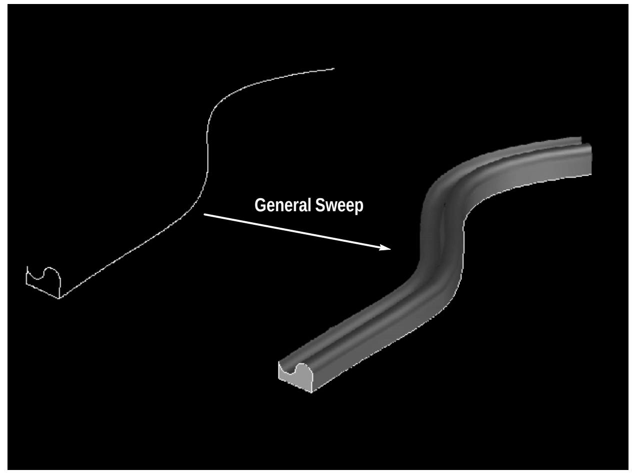


(a)

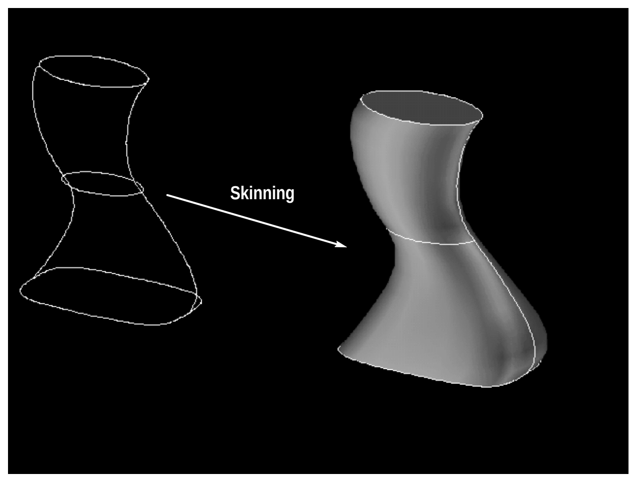


(b)

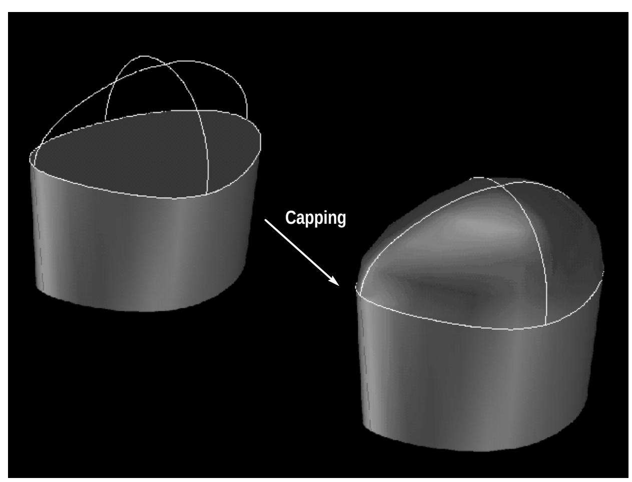
Fig. 18. (a) In creating a helical shape connected to a cylindrical shaft, if the helix base profile touches the cylinder a nonmanufacturable part results since the freeform helix oscillates around the cylinder surface. (b) If the helix base profile cuts into the cylinder a little the oscillating surface lies completely inside the cylinder and the unification of the two bodies will yield the expected result.



(a)



(b)



(c)

Fig. 19. (a) Sweeping. (b) Skinning. (c) Capping.

For this reason, one should never try to approximate other geometry using lofting in combination with a Boolean operation. It is much safer to create the loft tool body a little bigger to get clear intersections later. The example of Fig. 18 illustrates this. The task is to create a helical shape connected to a cylindrical shaft. The "workplane inclined" command is used to position the profiles for the loft. If the base profile touches the cylinder the unification of the lofted body and the cylinder will result in a nonmanufacturable part since the freeform helix oscillates around the cylinder surface (Fig. 18a).

However, if the profile cuts in a little the oscillating surface will lie completely inside the cylinder and the unification of both bodies will yield the expected result, as shown in Fig. 18b.

Summary

Lofting in HP PE/SolidDesigner is a powerful tool that enables the CAD user to create various freeform shapes within a solid model. The main task being solved by the user is the optimal selection of the profiles and clever positioning of the workplanes in the 3D space. With a little experience to gain familiarity with the behavior of the surface interpolation algorithms, many design tasks can be done in a short time. However, some tasks are cumbersome or nearly impossible using lofting, but are easily done using other HP PE/SolidDesigner capabilities. In electromechanical and mechanical engineering these tasks include mainly skinning, capping, and sweeping. Sweeping (Fig. 19a) is related to lofting since it means creating a surface by sweeping a profile along an arbitrary 3D curve. Skinning (Fig. 19b) is the task of defining a smooth surface through a net of 3D curves. Capping (Fig. 19c) means the replacement of a closed loop on a body by some smooth, tangentially connected surface; it is a subclass of skinning. Although these functionalities are the classical domain of surface modeling systems the open architecture of HP PE/SolidDesigner readily accommodates their implementation.

References

1. 1. G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*, Academic Press, 1988.

2. C. deBoor, *A Practical Guide to Splines*, Applied Mathematical Sciences no. 27, Springer, 1978.