Expanding the Use of Solid Modeling Throughout The Engineering Curriculum

Douglas H. Baxter

Rensselaer Polytechnic Institute

Abstract

The use of the studio classroom at Rensselaer Polytechnic Institute has greatly enhanced the traditional classroom experience. With computers, students are now able to work with their instructors through example problems. One of the added benefits of the studio classroom is that students can enhance their skills using the various software packages available to them for their work. As a prime example, with the visual capabilities of today's solid modeling packages, solid modeling examples can be used to enhance the students' abilities to grasp fundamental engineering concepts in their studies. The effective use of solid modeling for design and documentation of parts and assemblies is well established. In this paper, uses for solid modeling outside of traditional graphics courses are examined. It is shown how solid modeling can be used to visualize concepts taught in undergraduate studies; including vector properties, differentiation as applied to shape optimization problems, and simple structural optimization. While the focus of this paper will be on developing visualization skills at the freshman and sophomore levels, it will can seen how solid modeling can be employed throughout a four-year program. By taking advantage of the computing power available to students in the studio classroom, key concepts can be visualized with the aid of solid models.

Introduction

The use of computers in engineering education has continued to increase. Computers were once the subject of courses, and were also used as one of the many tools engineering students needed in their course work. Today, computers have become major teaching tools, the dominant tool for the engineering student. At Rensselaer, incoming students are now required to purchase laptop computers to a minimum performance specification. Many of the freshman courses now use the laptop computers in the classroom; the most common type being the studio classroom. Such studio classes will propagate throughout the curriculum as the present freshman class progresses. Traditional lecture courses will continue to be replaced with studio courses. Those involved in teaching the present freshman studio courses are examining how students utilize their laptop computers within and outside of the classroom in order to better develop subsequent studio courses.

One conclusion from teaching studio courses is that students must quickly master the use of their laptop computers and how they work with the computing system at Rensselaer. The students computers are connected to the Rensselaer system in their classroom via an ethernet connection. Ethernet connections have been installed throughout the campus in major study areas (such as the library and student lounges) and residence halls as well as in many of the classrooms on campus. Students are encouraged to use the ethernet connections for data storage and retrieval, printing and access to course materials. The campus computing center has worked to minimize the impact of moving data between the laptop and the computing system. Despite their efforts, it is still necessary to include course material that emphasizes computer skills. This is a concern as these specific skills are necessary to use computers at Rensselaer but may not be as useful elsewhere. Furthermore, time spent teaching specific skills takes time away from teaching fundamental principles.

The balance between teaching skills and fundamental principles is further strained by making students learn several major software packages during their freshman year. Students are expected to use these software packages throughout their career at Rensselaer. Students must understand how to use the software, and, more importantly, how the software works. If students have some understanding of what the software is accomplishing, then they are better able to recognize when the software can be utilized in future assignments. Again, it becomes necessary to teach a skill (using the software) but more importantly, to teach the fundamental principles so that students understand why the skill they are learning applies to the problem. The balance between these two needs must be constantly monitored. An excellent example of this balance is in the required (for all engineering majors) freshman computer graphics course, Engineering Graphics and Computer Aided Design (EG&CAD).

EG&CAD requires engineering students to learn the fundamentals of solid modeling and visualization by creating and documenting solid models of parts and assemblies. The fundamentals of solid modeling are taught by using vector analysis. This provides students insight into the workings of the software and also provides a practical application to the linear algebra work they are doing in other courses. In addition, the students are required to do some limited hand sketching as several studies have indicated that hand sketching improves visualization skills [9]. All engineering students at Rensselaer are required to take EG&CAD. Students will be required to use their solid modeling skills in their subsequent design courses where documentation of their designs is required. For most of the engineering students, the next opportunity to use solid modeling is in their sophomore year for Introduction to Engineering Design (IED). Many students in IED find that documenting their design is difficult as they have not used a solid modeler in several months. Valuable design time is lost as students must reacquaint themselves with the software.

It had been hoped that a studio course format for EG&CAD would help students better retain their solid modeling skills while increasing the time spent on learning fundamentals. EG&CAD became a studio course in 1996 [5] using campus UNIX terminals. The course switched to Windows desktop systems in 1999. The studio format has increased the amount of fundamental material presented in the course [2] and [3] but the students' retention of solid modeling skills has not appreciably increased.

Proposal

To increase the students' retention of solid modeling skills and understanding of fundamental principles, it has been proposed that teaching aids using solid modeling be developed for basic engineering courses common to most engineering students. By utilizing the visualization capabilities of a solid modeling system, it is hoped that students will be able to better understand basic concepts taught in their freshman and sophomore courses while having more opportunities to use the solid modeling system outside of the course work in EG&CAD. As these examples are meant to be used by students with their laptop computers, care was taken to ensure that the software delivered with the laptop (plus the solid modeling program) would suffice to run the example problems. Three classroom examples for three different courses will be presented to show how the visualization capabilities of a solid modeling system may be used outside of a computer graphics course. Two of the examples have been used on a trial basis in the basic engineering courses for which they developed. The third example has been developed for an upper level course in design optimization, and has not yet been used in the classroom.

Vector Addition Example[1]

The first example was developed for Introduction to Engineering Analysis (IEA). IEA is an introductory course in linear algebra using engineering examples. The course is required for all first year engineering students. Many students take IEA and EG&CAD in the same semester, although some will not take EG&CAD until a semester after they have taken IEA. Vector addition, first in two dimensions, and then three dimensions, is covered in the beginning of the course. To aid in visualizing the properties of orthogonal vectors, a small solid model was developed as shown in Figure 1.

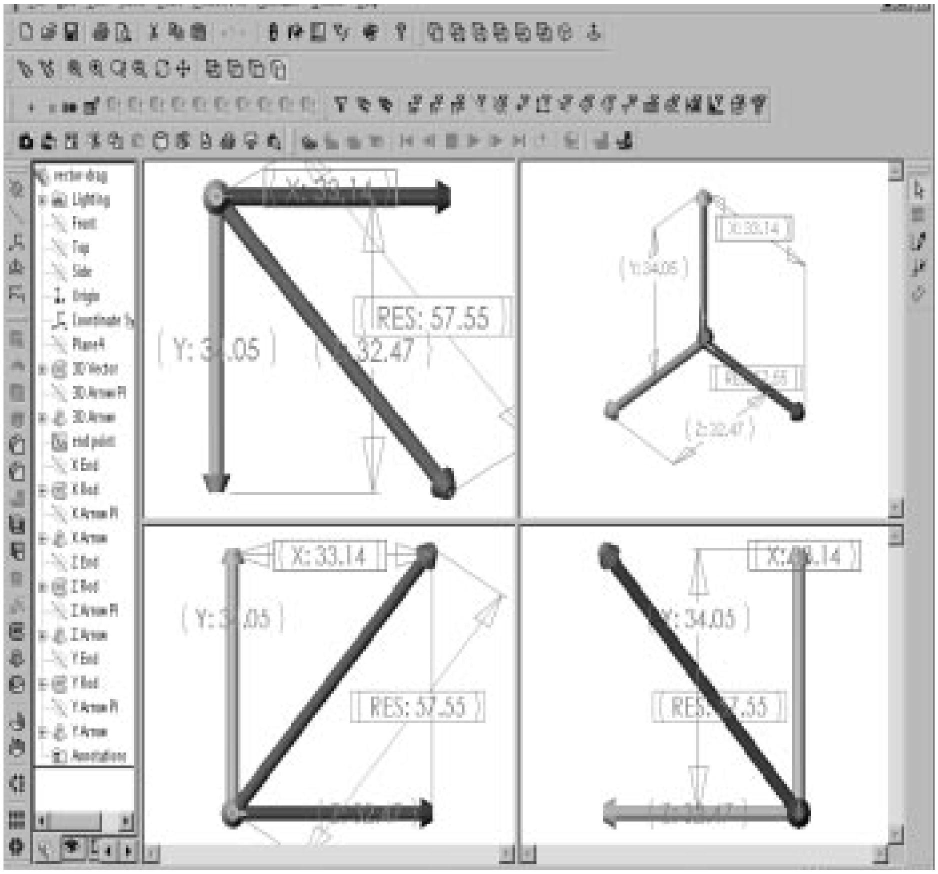
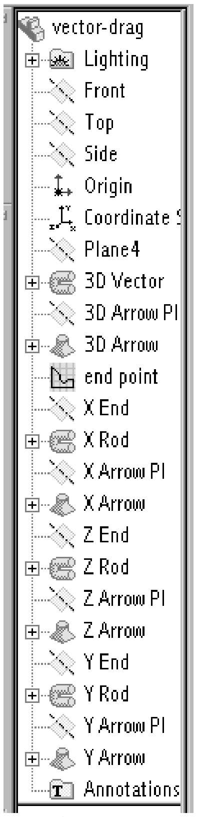


Figure 1 -Vector Addition Example

[1] All example files are available on request. To run the examples, the user must have Microsoft Office 2000 and SolidWorks 1999 at level 313 or above.

In Figure 1, four views of the vector addition SolidWorks solid model are shown. In the isometric view (top right) the X, Y, and Z vectors are clearly seen with the tip of the resultant vector seen at the intersection of the X, Y, and Z vectors. The three remaining views are the XZ (or top) view (top left), the XY (front) view (bottom left) and the ZY (right) view (bottom right). In each view, it can be seen that the resultant vector as viewed from the principal plane is the vector sum of the two vectors in the viewing plane.

The SolidWorks model was built without any dimensional constraints. The resultant vector was built by creating a three-dimensional line anchored at the origin. A circular cross-section was swept along the three-dimensional line. Finally, an arrow tip was created on the free end of the vector to give the resultant shape the appearance of an arrow. The remaining axial vectors were built as two-dimensional lines starting at the origin and running along a principal axis. The ends of the two-dimensional lines were aligned with the end point of the resultant vector as viewed from the normal plane of the two-dimensional lines. The model tree for the solid model is shown in Figure 2. Note that the features have been labeled so that students can easily identify both the solid portions of the model and the construction geometry (datum planes and curves) used to create the solid model.



vector-drag
Lighting
Front
Top
Side
Origin
Coordinate S
Plane4
3D Vector
3D Arrow Pl
3D Arrow
end point
X End
X Rod
X Arrow Pl
X Arrow
Z End
Z Rod
Z Arrow Pl
Z Arrow
Y End
Y Rod
Y Arrow Pl
Y Arrow
Annotations

Figure 2 -Model Tree forVector Addition Example

The student can select the resultant vector and then modify the length of the resultant vector by pulling on the end point of the resultant vector. By using a click and drag operation, the student can move the end of the resultant vector. The new length of the vector is shown and the three principal axis vectors are automatically updated and their new lengths are also shown. An example of a modified resultant vector is shown in Figure 3.

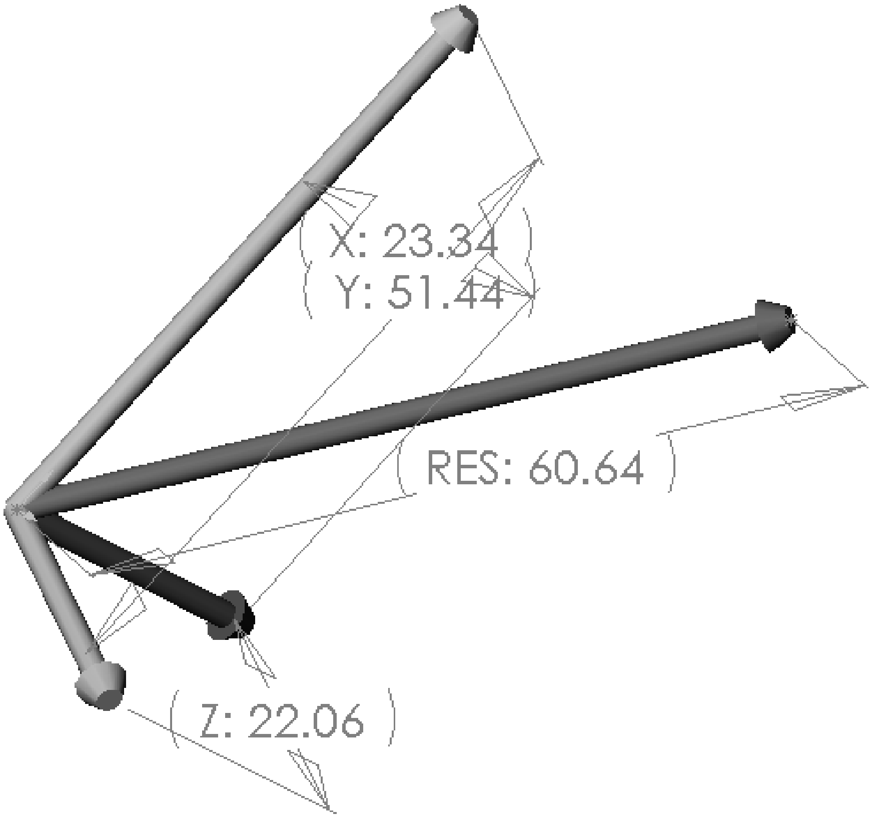


Figure 3 -New ResultantVector

Note that the three dimensional view shown in Figure 3 is not an isometric view. One of the advantages of using a solid modeling system is that the student can rotate the solid in any orientation for viewing. The six principal orthographic views and the isometric views are available via icons but any view can be created and then saved if desired. Only minimal knowledge of the solid modeling software is necessary to run this example; the student needs to know how to manipulate the view and how to modify the part size with stretch and drag commands. As these commands are similar to other Windows programs, most students have been able to quickly master these simple commands. Further exploration of the solid model can lead to a better understanding of how to build constrained features and how to use existing geometry in the creation of new geometry within a solid model.

Shape Optimization Example

The second example is taken for a IED homework problem that asks students to maximize the volume of an open ended right cone given a maximum surface area of 200 in^2. Students must document their answer (including references for the equations) and prove that they have found a true maximum. The problem is a simple calculus problem that is easily solved once the formulae for the cone's area and volume are known. Most of the students solve this problem by using the area equation to reduce the volume equation to one variable (either the cone height or cone base diameter), and then differentiate the volume equation twice to solve for the cone height and the cone base diameter and to ensure the solution is a maximum. As the problem is very simple, the goal of this homework assignment is to get the students to practice proper documentation of their analysis and to consider how mathematics is applied to engineering problems. With this example, it is possible to introduce design optimization as a means to solve this problem. To help visualize the problem, a surface model of the cone was created as shown in Figure 4.

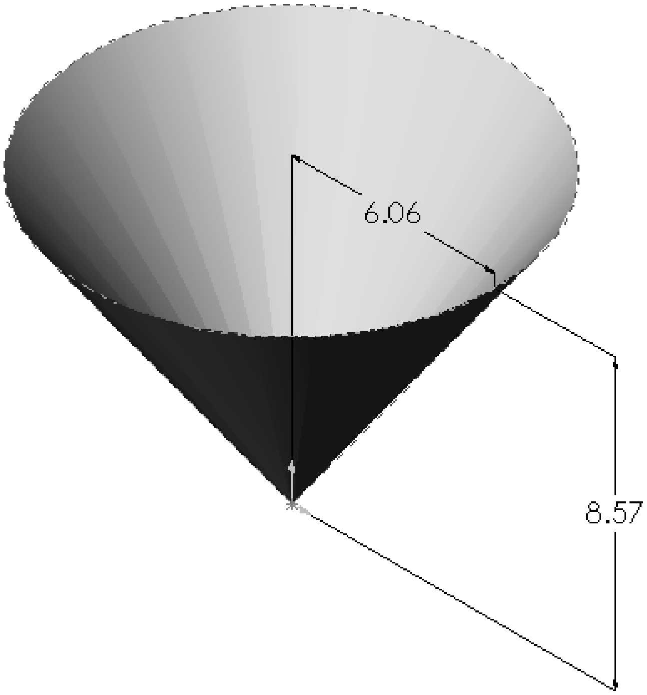
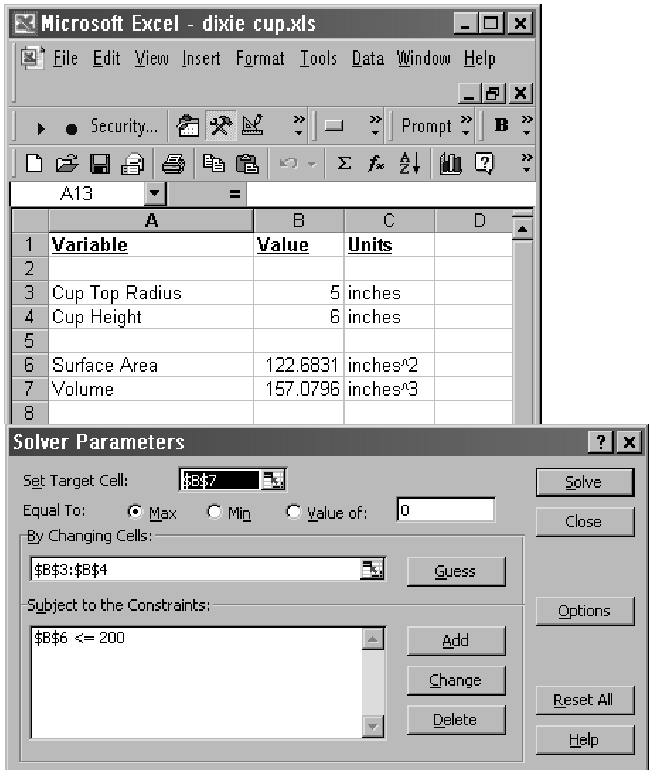


Figure 4 - Surface Model of Open Ended Right Cone

Students have three software packages that can be used to create a design optimization model to solve this problem. Microsoft Excel was used as Excel's solver is very easy to use. The other two packages (Maple and MatLab) would require writing the subroutines to solve the problem. The Excel spreadsheet is shown in Figure 5. The variables of the problem are the cone base diameter and the height. The surface area and volume cells are formulae taken from [8]. The solver window indicates the goal is to maximize the volume (cell B7) by varying the cone height and cone base diameter (cells B4 and B3). A constraint equation states that the surface area (cell B6) is less than or equal to 200 in^2.



	A	B	C	D
1	Variable	Value	Units	
2				
3	Cup Top Radius	5	inches	
4	Cup Height	6	inches	
5				
6	Surface Area	122.6831	inches^2	
7	Volume	157.0796	inches^3	
8				

Set Target Cell: $B$7

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells: $B$3:$B$4

Subject to the Constraints: $B$6 <= 200

Figure 5 -Spreadsheet For Cone Volume Maximization

Using Excel's solver, a solution can quickly be reached. Excel uses a gradient solver. To visualize how the solver converges on a solution, Visual Basic is used to link the Excel program with the solid modeler via Microsoft Windows API (**A**dvanced **P**rogram **I**nterface). Visual C++ could also be used but the Visual Basic is supplied with Microsoft Office and is thus readily available to the students. The Visual Basic code is shown in Figure 6. The code is executed when a change to the Excel spreadsheet is made. The code was written by using the macro editor in Excel and specifying how the macro is executed (by change of variable). Linking the cells in Excel and the dimension values in the SolidWorks model are done with a single line of code for each variable. The rest of the Visual Basic Code is used to set the units in Excel by reading the units from SolidWorks. The end of the code rebuilds the SolidWorks model and fits the model within the screen. The macro does not need to be as long as the macro shown in Figure 6, but as this is a student example, care was taken to put in safety checks and comments to ensure that the macro is easy to read.

```
Public swApp, Model, Part As Object
Public Busy As Boolean
Public Conversion AsVariant
Public retval, result As Long
Private Sub Worksheet_Change(ByVal Target As Range)  ‘Changes in the spreadsheet start the macro
  If Not Busy Then
  'Disable this code from change event while it is running
  Busy = True
  Set swApp = CreateObject("SldWorks.Application")
  Set Model = swApp.ActiveDoc
  If (Model Is Nothing) Then                    ' No SolidWorks Model
   MsgBox ("Open A SolidWorks Model To Begin")
  Exit Sub
 End If
 units = Model.GetUnits                         ' Get units of Assembly
 If (units(0) = 0) Then                        ' Units are mm
  Conversion = 1000
  Range("C3") = "mm"
  Range("C4") = "mm"
  Range("C6") = "mm^2"
  Range("C7") = "mm^3"
 End If
 If (units(0) = 1) Then                        ' Units are cm
  Conversion = 100
  Range("C3") = "cm"
  Range("C4") = "cm"
  Range("C6") = "cm^2"
  Range("C7") = "cm^3"
 End If
 If (units(0) = 2) Then                        'Units are meters
  Conversion = 1
  Range("C3") = "m"
  Range("C4") = "m"
  Range("C6") = "m^2"
  Range("C7") = "m^3"
 End If
 If (units(0) = 3) Then                        'Units are inches
  Conversion = 12 / 0.3048
  Range("C3") = "inches"
  Range("C4") = "inches"
  Range("C6") = "inches^2"
  Range("C7") = "inches^3"
 End If
 If (units(0) = 4) Then                        'Units are feet
  Conversion = 1 / 0.3048
  Range("C3") = "feet"
  Range("C4") = "feet"
  Range("C6") = "feet^2"
  Range("C7") = "feet^3"
 End If
 'Cup Radius
  Model.Parameter("Cup Radius@Sketch1").SystemValue = Sheet1.Range("B3") / Conversion
  'Cup Height
  Model.Parameter("Cup Height@Sketch1").SystemValue = Sheet1.Range("B4") / Conversion
  'Rebuild the part with new dimensions
  Model.EditRebuild
  'Resize part to fit the screen
  Model.ViewZoomtofit2
  'Enable this code for next change
  Busy = False
  End If
End Sub
```

Figure 6 -Visual Basic Code For Cone Optimization

When the solver in Excel is executed, the student can watch the SolidWorks model vary until a solution is reached. The student can see how the variables change and how solver tends to vary one variable at a time usually overshooting the desired value and then returning to the optimal value. What was a simple calculus problem becomes a visual exploration of the design space. Furthermore, the student can see that analysis tools can be used to read and send data to the solid model so that the final design parameters are entered by the computer instead of the student having to update the solid model. The student is introduced to the operating system's programming interface, and sees how different programs can be linked together to form powerful engineering tools. There is a considerable amount of Visual Basic subroutines available from Microsoft and SolidWorks [6]. Given the amount of help available, it is not difficult to learn how to use the Visual Basic API.

This problem can be expanded into a three variable problem by replacing the right circular cone with a frustum of right circular cone as shown in Figure 7. This problem does not have a closed form solution given the constraints of the original problem. The modification to the Visual Basic macro is one additional line of code to call the new dimension (the smaller diameter in Figure 7), and an additional line for each of the units selection. The area and volume cells must be modified and an additional constraint to keep the bottom diameter less than or equal to the top diameter of the frustum of cone must be added.

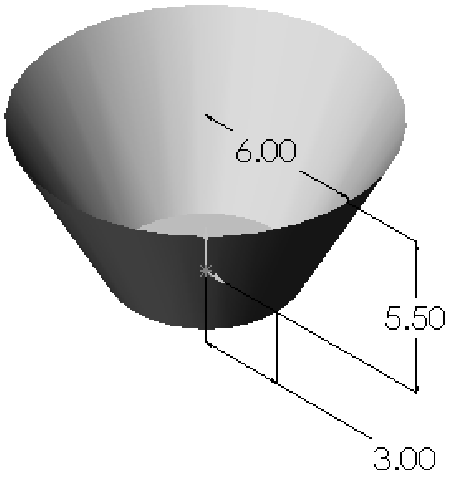


Figure 7 -Solid Model of a Frustum of Right Circular Cone

Four Variable Design Optimization Problem: Welded Beam Example

The welded beam example is a classic design optimization example and is well documented [7]. With four variables, it is difficult enough that a graphics solution is not practical, yet the formulation of the bending and buckling equations are not difficult. This problem was written for the undergraduate design optimization course. This course is normally a senior-level course. The problem, as shown in Figure 8, consists of a beam welded on to a plate with a point source load at the free end of the beam. The goal of the optimization is to minimize the cost of the structure as constrained by maximum allowable deflection, stress, shear, and buckling. The cost of the structure consists of the cost of the beam (material cost only), the cost of the weld material, and the cost to apply the weld. The four design variables are the cross sectional dimensions of the beam (height and thickness), the weld width (the weld is assumed to be an right isosceles triangle), and finally the length of the weld. The Excel spreadsheet is shown in Figure 9, and the solver setup is shown in Figure 10.

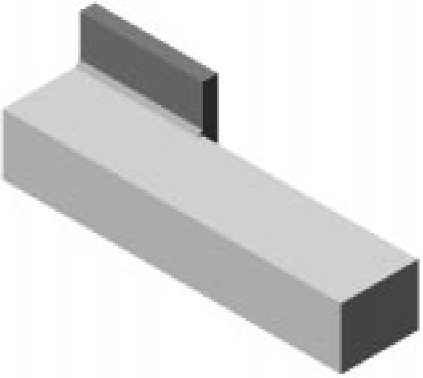
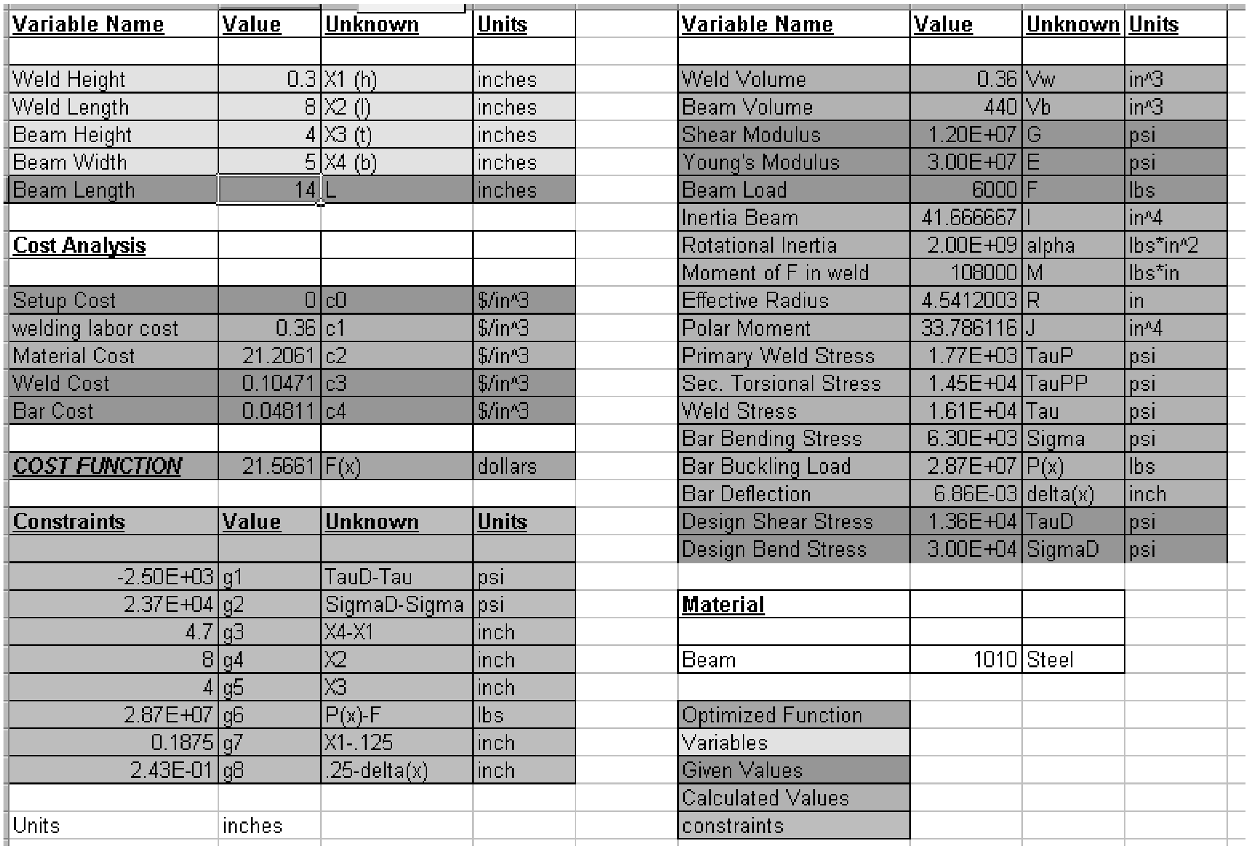


Figure 8 -Welded Beam Assembly

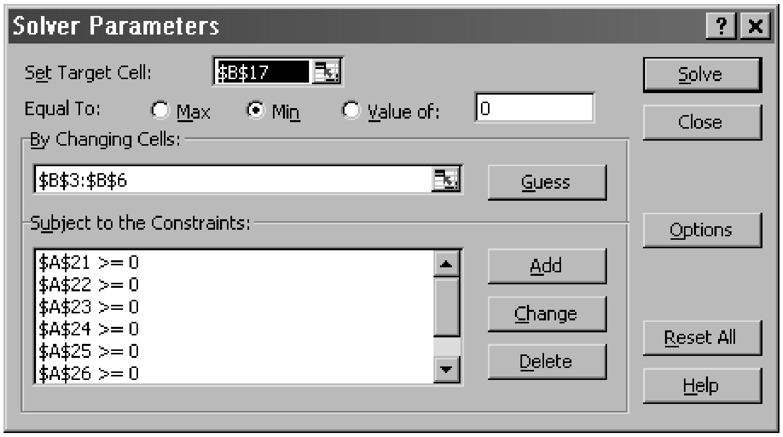
The macro for the welded beam example is very similar to the right circular cone example macro in Figure 6. One advantage of using the Microsoft API is that the equations are set up in Excel and the macro just feeds the data back and forth between Excel and SolidWorks. This allows the macros to be stored as templates and retrieved and modified for each example.



Variable Name	**Value**	**Unknown**	**Units**
Weld Height	0.3	X1 (h)	inches
Weld Length	8	X2 (l)	inches
Beam Height	4	X3 (t)	inches
Beam Width	5	X4 (b)	inches
Beam Length	14	L	inches
Cost Analysis			
Setup Cost	0	c0	$/in^3
welding labor cost	0.36	c1	$/in^3
Material Cost	21.2061	c2	$/in^3
Weld Cost	0.10471	c3	$/in^3
Bar Cost	0.04811	c4	$/in^3
COST FUNCTION	21.5661	F(x)	dollars
Constraints	**Value**	**Unknown**	**Units**
-2.50E+03	g1	TauD-Tau	psi
2.37E+04	g2	SigmaD-Sigma	psi
4.7	g3	X4-X1	inch
8	g4	X2	inch
4	g5	X3	inch
2.87E+07	g6	P(x)-F	lbs
0.1875	g7	X1-.125	inch
2.43E-01	g8	.25-delta(x)	inch
Units	inches		

Variable Name	**Value**	**Unknown**	**Units**
Weld Volume	0.36	Vw	in^3
Beam Volume	440	Vb	in^3
Shear Modulus	1.20E+07	G	psi
Young's Modulus	3.00E+07	E	psi
Beam Load	6000	F	lbs
Inertia Beam	41.666667	I	in^4
Rotational Inertia	2.00E+09	alpha	lbs*in^2
Moment of F in weld	108000	M	lbs*in
Effective Radius	4.5412003	R	in
Polar Moment	33.786116	J	in^4
Primary Weld Stress	1.77E+03	TauP	psi
Sec. Torsional Stress	1.45E+04	TauPP	psi
Weld Stress	1.61E+04	Tau	psi
Bar Bending Stress	6.30E+03	Sigma	psi
Bar Buckling Load	2.87E+07	P(x)	lbs
Bar Deflection	6.86E-03	delta(x)	inch
Design Shear Stress	1.36E+04	TauD	psi
Design Bend Stress	3.00E+04	SigmaD	psi
Material			
Beam	1010	Steel	
Optimized Function			
Variables			
Given Values			
Calculated Values			
constraints			

Figure 9 - Excel Spreadsheet for the Welded Beam Example


Solver Parameters

Set Target Cell: $B$17

Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: $B$3:$B$6

Subject to the Constraints:

```
$A$21 >= 0
$A$22 >= 0
$A$23 >= 0
$A$24 >= 0
$A$25 >= 0
$A$26 >= 0
```

Figure 10 -Excel Solver Setup for Welded Beam Example

The advantage of using the macros to visualize the optimization is that students can see how the optimization code behaves. The deflection constraints can be modeled with datum planes displayed in the SolidWorks session. The color of the datum plane can indicate if the constraint is active or inactive. Similarly, the color of the parts can be changed to reflect the status of the other constraints. The final shape for this example is a long slender beam as shown in Figure 11.

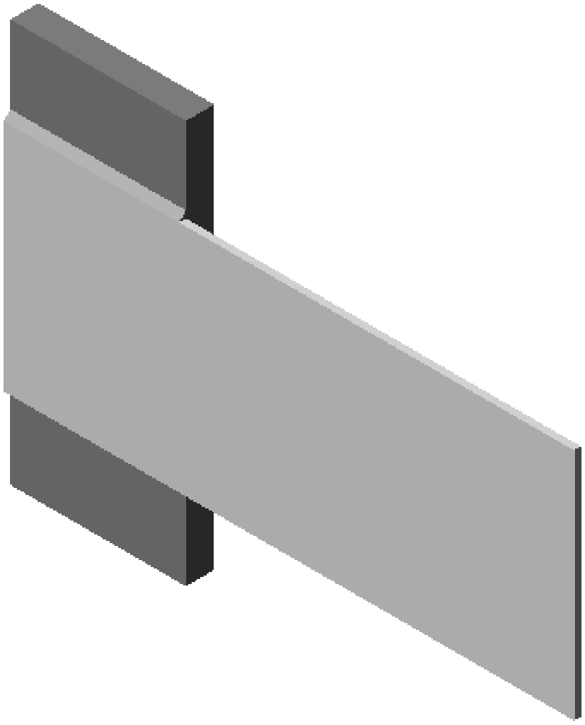


Figure 11 -Final Beam Shape

Future Work

Future work on these examples is planned. The welded beam example could be greatly enhanced by using different solvers. Immediate plans are to implement a simulated annealing solver to show how this class of solver differs from a gradient solver. Additional IED homework problems could be modeled and there are plans to add two new homework assignments next semester. Finally, there may be an opportunity to use the solid modeling software in other courses. Structures, dynamics, fluid mechanics, heat transfer and mechatronics should benefit from these types of examples. New software will be needed to make examples for these courses worthwhile.

These examples and the new examples in development will be used in classes over the next semester with the hope that a measurable improvement in students' visualization skills can be measured. A visualization quiz has been developed to measure students' progress. Pending the results of the testing, the examples will be incorporated into the courses for which they were developed.

Concluding Remarks

Solid modeling software can be used for more than computer graphics courses and follow on design work. The visualization capabilities coupled with an open API allow for the integration of several engineering packages to demonstrate fundamental principles using the visualization abilities of the solid modeler. Care has been taken to develop models that present these fundamental concepts in a consistent manner with industry practices [4]. The models also follow course materials for EG&CAD [1]. The examples do not require students to modify the environment on their laptop computers. The goal is to have examples that can be run by students' outside of lecture with minimal instructons. Engineers need to be able to visualize the problems they are trying to formulate and solve. Engineering students also have the need to learn new tools and understand how these tools are applied to engineering problems. By providing the students examples that use the software they will be using in their course work, the students have more exposure to these software packages and thus increase their skills in using these software packages.

Bibliography

1. Baxter, Douglas, and Bunk, Donald, *Engineering Graphics and Computer Aided Design*, Primus Custom Publishing, McGraw-Hill, New York, New York, 1999.

2. Baxter, Douglas, "Comparing Passive and Active Lectures for a Freshman Computer Aided Design Course," *Proceedings, American Society of Engineering Education: Engineering Design Graphics Division*, Columbus, Ohio, January, 1999.

3. Baxter, Douglas, "Engineering Graphics and Computer Aided Design: A Foundation to Engineering Design and Analysis", *Proceedings, American Society of Engineering Education: Annual Meeting,* Seattle, Washington, June, 1998.

4. Bertoline, Gary, Wiebe, Eric, Miller, Craig, and Mohler, James, *Technical Graphics Communication, 2nd Edition*, WCB McGraw Hill, Boston, Massachusetts, 1997.

5. Bunk, Donald and Baxter, Douglas "An Online Course In Solid Modeling", *Proceedings, American Society of Engineering Education: Regional Meeting,* Union College, Schenectady, New York, September, 1995.

6. New Hampshire CAD SolidWorks API Tutorials, http://www.nhcad.com, available as of December 1, 1999.

7. Reklaitis, G., Ravindran, A., and Ragsdell, K., *Engineering Optimization Methods and Applications*, John Wiley Sons, NewYork, NewYork, 1983.

8.*Standard Mathematical Tables, 25th Edition*, William H. Beyer, Editor, CRC Press, Boca Raton, FL, 1980.

9. Woolsey, Kristina, Kim, Scott, Curtis, Gayle, *VizAbility*, PWS Publishing, Boston, Massachusetts, 1996.

DOUGLAS BAXTER

Douglas H. Baxter is a Lecturer in Core Engineering at Rensselaer Polytechnic Institute in Troy, New York. In addition to being the coordinator of EG&CAD and teaching IED, He is presently completing his doctoral work in Mechanical Engineering. He received his B.S.M.E. from Syracuse University in 1983 and his M.S.M.E. from Rensselaer Polytechnic Institute in 1987. Prior to joining RPI, he was a design engineer at IBM.