

# A COMPUTER SOLAR ANALYSIS TOOL FOR THE DESIGN AND MANUFACTURING OF COMPLEX ARCHITECTURAL ENVELOPES: EVSURF

Jose da Veiga

Architect, MA  
MA student - Department of Architecture at the  
University of California Los Angeles  
jfernand@ucla.edu

Pablo La Roche

Architect, MSc, Professor of Architecture at the  
University of Zulia in Venezuela –  
PhD candidate - Department of Architecture at the  
University of California Los Angeles  
plaroche@ucla.edu

## Resumen

Los flujos de calor en una edificación son afectados por la radiación solar, las ganancias internas y las diferencias de temperatura. Se ha desarrollado en formato applet, un prototipo de herramienta que evalúa la radiación solar en envolventes complejas de edificaciones en función de la cantidad potencial de radiación que puede recibir la superficie. Una función de fitness evalúa el desempeño de la edificación permitiendo comparar alternativas en diferentes días y horas. Debido a sus características esta es una herramienta de diseño automatizada muy sencilla que puede ayudar al arquitecto a diseñar edificios más eficientes energéticamente.

## Abstract

Heat flows (gains or losses) in a building are affected by solar radiation, internal gains and temperature differences. A prototype of a tool that evaluates solar radiation on building envelopes has been developed as an applet to run on the Internet. This tool analyses the relationship of the solar position with the surfaces of complex envelopes by evaluating the potential amount of direct solar radiation that each surface section receives as a function of the angle of each surface with the sun for specific days and latitudes. A fitness function rates the overall performance of the proposed building envelope at specific hours or whole days, permitting to compare alternatives. Because of its characteristics, this tool is a Very Simple Design tool that works interactively with the user during the design process helping to design solar responsive building envelopes.

## Introduction

This paper presents a Web based prototype for a computer program that helps the designer in determining the effects of direct solar radiation in complex building envelopes. This is achieved by a visual representation of a quantitative ratio and rating between the possible solar radiation values with the surface's area throughout the day. The prototype has a strong graphical component and is can be used as an analysis tool during the design process, especially during the initial phases, where graphic visualization of results is more important than numerical results.

## Solar Algorithms

Well-known equations for obtaining solar position permit to predict the position of the sun in relation to any point in the earth at any time and day of the year. For our purposes, the only inputs required for these calculations are the latitude and the number of the day of the year. Any given exposed surface receives direct, diffuse and reflected radiation. The amount of radiation varies depending on many factors. In very cloudy climates the most important component is diffuse radiation, under very clear skies the most important component is direct radiation, and if the surfaces surrounding each component are reflective then the reflected component might be the most important. All of these components are affected by the angular relationship of each individual surface with the sun, the sky or the neighbouring surfaces. In "typical" conditions, direct solar radiation is the most important component of the global radiation.

Therefore this prototype only calculates the effect of the direct component of solar radiation. The diffuse and reflected components are considered fixed in each surface. The potential for radiation is expressed as a function of the angle of the solar rays with the building element. Five levels of intensity are proposed in a scale that divides the maximum possible value (incident angle of zero) in five equal parts. Since the direct radiation level on a plane is a function of the angle of incidence of the sun's rays with the normal to the plane and the values that are generated are a function of the cosine curve, the values can be translated to a percentage of the maximum radiation possible, as a function of the cosine of the angle. Figure 1 shows how the angular distance relates to the percentage scale of maximum possible direct solar radiation. The numbers do not indicate quantitative radiation values but rather are expressed as a percentage of the maximum possible number (figure 1).

Angle to Normal	Percentage of Total	Intensity Scale	Color Shade
$0^\circ \rightarrow \alpha < 37^\circ$	$100 \rightarrow \alpha < 99$	1	Yellow
$37^\circ \rightarrow \alpha < 54^\circ$	$69 \rightarrow \alpha < 69$	2	Orange
$54^\circ \rightarrow \alpha < 67^\circ$	$48 \rightarrow \alpha < 48$	3	Red
$67^\circ \rightarrow \alpha < 79^\circ$	$28 \rightarrow \alpha < 28$	4	Green
$79^\circ \rightarrow \alpha < 90^\circ$	$0 \rightarrow \alpha < 0$	5	Black

Fig 1 - Table - Radiation Scales used in the graphics

To be able to numerically compare surfaces with each other, and to evolve surfaces from initial envelope proposals, there has to be a fitness function. This fitness function compares hourly daily values

and averages. The number is calculated as a function of the relationship of the intensity in each area. For each hour of the day in which the sun is above the horizon, the program calculates a weighed average of the performance of all the surfaces so as to be able to determine the hourly performance of the total envelope (Rittel, 1970):

- (1)  $AHP = ((SA1 * NS1) + (SA2 * NS2) + (SA3 * NS3)) / NSurf$   
 AHP= Average Performance for each Hour  
 SA= Surface area  
 NS1= Percentage or number from scale (e.g. 1,2,3...)  
 NSurf = Number of surfaces

Then the weighted daily performance can be determined by calculating the average of the hourly averages for the daylight hours. This would be the performance of that envelope for that specific day.

- (2)  $DAP = ((AHP 1 + AHP 2 + AHP 3 + \dots)) / NHD$   
 DAP= Diurnal Average Performance  
 AHP= Average Performance for each Hour  
 NHD= Number of Hours of Daylight

The AHP value permits to determine the performance of an envelope in any given hour and day as a snapshot, while the DAP value permits to evaluate the performance of the envelope in specific days. The analysis of days such as the solstices or the equinoxes will permit to determine the performance of the building envelope during the year.

### “EvSurf” as a Computer Aided Design Tool

A CAD tool can be any computer program that helps the architect to solve an issue during the architectural design process. The ultimate objective of all CAD tools is to improve the product by improving the design process (Rittel 1970,). The importance of shading as a design strategy to reduce overheating in buildings has been recognized for a long time (Olgyay, 1957), (La Roche, 2001) and the algorithms that define solar position in the sky vault are well known (Szokolay, 1996). This combination has led to the development of many computer tools to analyse the effects of shading in the architectural and urban scale. Some have been developed to aid in the design of shading devices, such as overhangs, in the façade as stand alone programs (Han, 1986), (Setiadarma, 1996), (Bryan, 2001), or as part of more complex energy modelling programs (Milne, 2001) and other have been developed to aid in the design of buildings and their open spaces, considering shading or solar rights in the urban context, (Yeziro, 1994), (Knowles, 2000), (Schiler, 1993).

Little has been done to determine the effect of radiation in building envelopes with complex surfaces in different orientations, such as those that can be generated with current 3D modelling software. On the design field, there have been many advances in the development of software that permits to model complex buildings. The use of this software from design to construction has already generated many projects with complex surfaces and forms, such as the Guggenheim Bilbao Museum by Frank Gehry. There has been no similar advance in solar design tools in which the gap between the generation and the analysis of the form persists.

Many newer energy design tools are now available for download or direct use through the Internet. Examples of this are the URL of Sustainable by Design (Gronbeck, 2000), the Department of Architecture and Urban Design in the University of California Los

Angeles (Milne, 2000), and the School of Architecture at the University of Southern California (Schiler, 1999). The importance of these web sites increases every day (Schiler, 2000) and these few examples have Design Tools that can be used directly from the web or downloaded free of charge.

### EvSurf as a Very Simple Design Tool

The focus of the prototype program presented here is to analyse the relationship of solar radiation with the building envelope so as to determine appropriate building features for specific locations as expressed by the latitude. This applet permits the designer to determine the performance of any type of building envelope by relating the solar position with the different surfaces and generating values that relate the calculated amount of direct solar radiation in the surface with the maximum possible calculated. The results are colour coded or grey scaled in the envelope and can be also expressed quantitatively.

According to La Roche (La Roche, 2001) Very Simple Design Tools, (VSDs) are very easy to use, making them accessible to a large base of non-experts in the fields of computation or sustainability; have a very short learning curve; and are accessible to a large number of users, the Internet being the preferred medium. On the other hand, these tools are not very precise, making them appropriate only for the early stages of the design process when there is no need for great precision. VSD tools should improve the quality of the architectural project by acting in the initial phases of the process, while at the same time complement the precision of tools that are used in more advanced stages of the design process.

### The Program

This program has been developed using HTML and Java. HTML provides instructions for the Client software on how the documents should be displayed and how to link to other documents on the Web. Java is an Object Oriented Programming Language that makes possible the creation of executable programs or applets that can be loaded in WWW browsers and are platform independent. This applet can be found at the following Internet address (Figure 2): <http://www.bol.ucla.edu/~jfernand/evsurf/index.html>

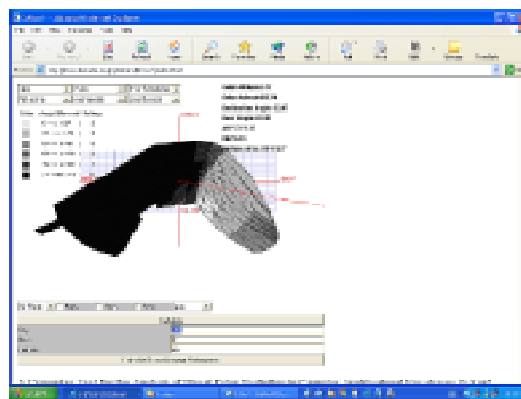


Fig 2 - Image of the Applet

This tool was built on algorithms on 3D environments in Java developed by Terzidis (Terzidis, 2002). The software is structured around a system that generates and represents a 3D virtual world to the user. To this representational system a calculation engine was added to provide for the sun angle calculations at each time of the day. The shading algorithms were adapted for the calculation of the sun's angle with each of the object's face. The objective is



not to develop the methodology for representing the shape but rather to elaborate a methodology of extracting information from the shape, analysing it and displaying that information in an intuitive manner.

By making use of the modularity provided by the object-oriented nature of the Java Programming Language we were able to create a structure to allow for the development of an independent set of objects to be utilized in the calculations. We accomplished this by creating a separation between the data representation engine (the 3D environment, coordinate system, etc) and the calculation engine responsible for evaluating the data (the object to analyse). We have developed a calculation engine for shape analysis that sits between the shape's data and the way the shape is represented; in this case the way it is rendered. The engine dictates the way the shape and the visual information are relayed to the user by controlling, in this case, the shading algorithm. This method will allow us to do further development by easily porting what we have learned into other more powerful CAD modelling and animation systems.

In figure 3 we can see a schematic representation of the program structure. Nr.1 on the figure is the overall program with the 3D environment, data structures, user interface, user input, etc. Nr. 2 represents the object data loaded from a DXF file. Nr. 3 is the user input done through the main program to be used by the calculation engine. Nr. 4 is where our algorithms for solar calculation. Nr. 5 is the rendering engine used to relay the information to the user. Normal 3D representation software functions without Nr. 4.

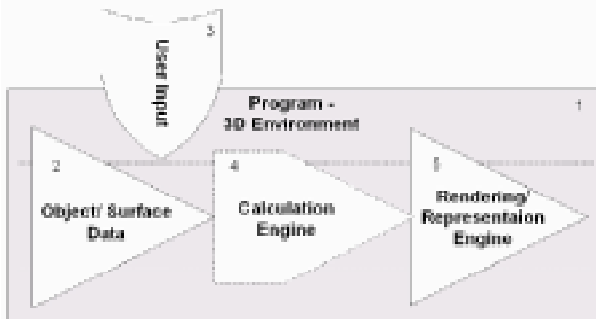


Fig 3 - Program Components

### Computer Algorithms

The objective of this program is to relate shape performance to exterior conditions (such as the sun) and represent the rating of that performance in an intuitive way. For the coordinate system we used a general rule for setting the location of the North, South, East and West cardinal points on the XYZ axis. In figure 4 you can see that the placement of the object in the xyz axis determines its orientation. The same happens for the sun. North corresponds to +Y, south to -Y, East to +X and West to -X. Z correspond to the altitude of the sun, - Z corresponds to night (no light).

The sun, a light vector, is correctly placed in the 3D environment by using the calculations described above. This allows for the environment to match the correct location of the sun based on time of the year, hour of the day and latitude. Using the calculations for the sun's position to obtain the angles of the sun with the North and with the XY plane, we derive the XY and Z utilizing simple sine and cosine calculations. This allows for a placement of the light representing the sun in our coordinate environment relative to the cardinal points and the object. Following the convention established for the coordinate axis in the calculations of the azimuth the Y value needs to be truncated to allow for a correct placement of the sun:

$$\begin{aligned}
 x &= \sin(\text{azi}); \\
 y &= -\cos(\text{azi}); \\
 z &= \sin(\text{alt});
 \end{aligned}$$

The only concern for the user is to create his/hers models using the same convention for a correct positioning of the shape to be analysed.

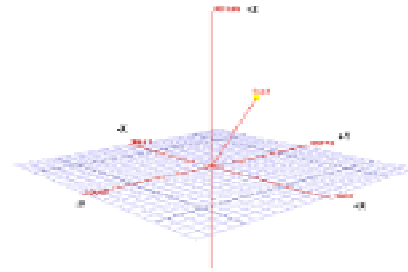


Fig 4 - Coordinate System

Changes were introduced to the shading algorithm for the correct representation of each object's performance. This is achieved by utilizing common shading algorithm and vector calculation procedures. As in shading a scale of grey is attributed to each surface component (a triangle) upon rendering (Terzidis, 2002). Usually this scale is defined by the angle of the light vector with the surface's normal. For our purpose a match of each component's performance is made with the grey scale. The correct number on the scale described in Figure 1 determines which shade of grey or colour the component will be painted with. The simplified pseudo code below showcases the simplicity of the pre-shading calculations for each face of the surface being analysed:

```

void shade(){
temp1.buildVector( points[0], points[1]
).cross(temp2.buildVector( points[1], points[2] ));
temp1.norm(); //normalize face vector created above
sun.light.norm(); //sun's vector and normalize it
visible = (int)( 90*temp1.dot(sun.light)); //calculate
angle of face normal with sun's vector
if (visible >= 79 && visible <90) { //if angle of face normal
with sun vector matches rating
faceColor = new Color(255,0,0); //rating colour to face
rating = 5; //rate face
}
...
...
if (sun.z <= 0) { //if sun below the line of the horizon, no rating
rating = 0;
faceColor = new Color(0,0,0); //face painted black.
}
}
  
```

The overall surface area is the sum of all the face's areas. For area calculation a simple algorithm was developed from a formula for the area of a triangle when all sides are known (Dildine, 1999). It is based on the triangle inequality theorem that states that the sum of two sides of a triangle must add up to be greater than the length of the remaining third side; where:

A triangle has sides a, b, and c.  
 After Calculating S, where  $S = (a+b+c)/2$   
 The Area of a Triangle =  $\text{SQRT}(s*(s-a)(s-b)(s-c))$



For calculating the size of each side of each face we used the very simple algorithm below based on creating a vector out of each pair of points of that same face:

```

v1 = (points[0].x - points[1].x);
v2 = (points[0].y - points[1].y);
v3 = (points[0].z - points[1].z);
v1 = v1*v1;
v2 = v2*v2;
v3 = v3*v3;
side1Lenght = sqrt(v1+v2+v3);

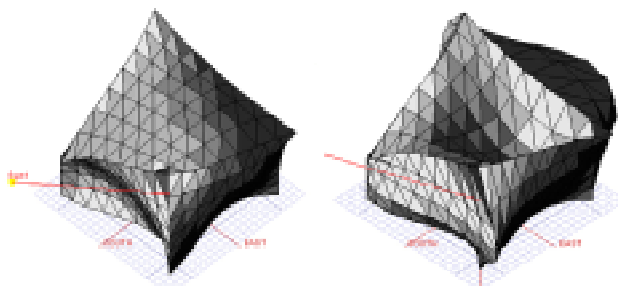
```

**“EvSurf” as an Analisis Tool:**

Two buildings with similar size are analysed and compared to illustrate potential uses of EvSurf. Building A (figure 5) has an area of 541556 ft², while building B (figure 6) has an area of 715.638 ft². Both buildings are evaluated at Latitude of 10N that could be a city such as Caracas, and 40 N, which could be a city such as New York. Three typical days are analysed to determine the overall performance: June 21 (day 172), March 21 (day 80) and DEC 21 (day 355). For each of these days and latitude a table with the DHP is shown. These figures give an idea of how the building performs for these three representative months and hours. For both buildings, the AHPs have different values depending on the time of day and date. The ideal situation for higher latitude would be to have higher ADPs in winter and lower in the summer. There ideal daily performances will depend on the rate of heat transfer of the envelope and the hours of use of the building.

Figures 5 and 6 are grey scale renderings of the effect of direct radiation at noon March 21 at Latitude 34 N. The program has the option for colour shading or grey scale renderings. Figures 7,8, 9 and 10 are the AHPs for 8 AM, Noon and 4 PM for figures 5 and 6. These permit to evaluate diurnal patterns for each surface during selected days of the year, in this case spring-fall, summer and winter. For latitude 10° the building has to reduce the solar radiation as much as possible during all the year while for 10° a high AHP is needed in winter and a low AHP in summer. A ratio is proposed that compares the summer to winter values as a ratio:

Winter AHP / Summer AHP = Seasonal AHP factor (SAHP)



	AHP 8:00	AHP 12:00	AHP 16:00	DAP
March	1528	1575	1312	1481
June	1471	1318	1248	1368
December	1486	1670	1340	1484

Fig 7 - Building A, 10 No LAT: AHP and DAP

	AHP 8:00	AHP 12:00	AHP 16:00	DAP
March	1479	1535	1288	1445
June	1527	1670	1343	1457
December	1384	1419	1288	1372

Fig 8 - Building A, 40 No LAT: AHP and DAP

	AHP 8:00	AHP 12:00	AHP 16:00	DAP
March	1857	1970	1685	1840
June	1928	1840	1605	1794
December	1672	2437	1657	1811

Fig 9 - Building B, 10 No LAT: AHP and DAP

	AHP 8:00	AHP 12:00	AHP 16:00	DAP
March	1698	1844	1637	1738
June	1948	2437	1698	1833
December	1397	1685	1558	1587

Fig 10 - Building B, 40 No LAT: AHP and DAP

The higher the value, the better the seasonal performance of the building envelope. If it is higher than one unit then the performance is even better. For these two buildings:

Building A: SAHP = 1372/1457 = 0.94

Building B: SAHP = 1587/1833 = 0.86

Thus, building envelope A is more effective on a seasonal basis.

To determine how the buildings compare with each other a normalized scale (units/area) can be used. Two options are possible, 1) using the area of the footprint of the building to determine the cost effectiveness per covered surface or 2) using the area of the envelope to determine the performance of the envelope itself. The first case would help to determine cost benefits of the surface area while in the second case it would permit to determine the cost effectiveness of the surface itself. The second option is selected and as an example, figure 11 (figure 11) compares buildings A and B in December 21 for Latitude 40°, at different hours.

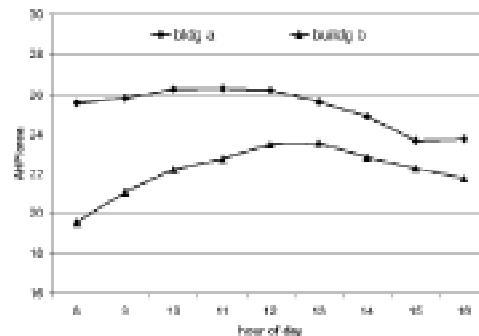


Fig 11 - Comparison of AHP between Building A and B for Lat 40 Dec 21.

Normalizing the data for each hour (SAHP) we can determine how the buildings perform during each hour of the day. For this particular day and latitude Building A is the better performer all the time, especially in the morning. Examples of the shading for building B are shown for December 21 at 8 AM, Noon and 4 PM at latitude 40° N (figure 12), (figure 13), (figure 14).





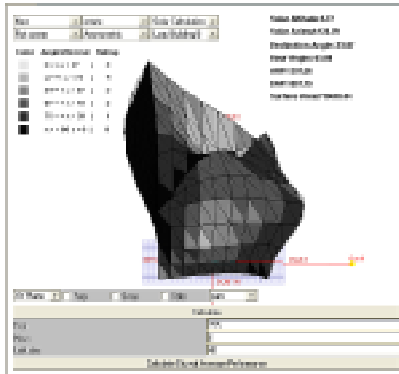


Fig 12 - Building B. Lat 40 N, Dec 21 at 8 AM.

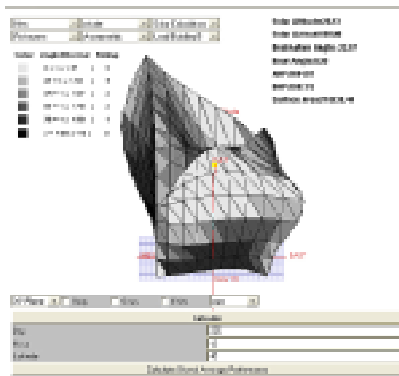


Fig 13 - Building B. Lat 40 N, Dec 21 at 12 noon

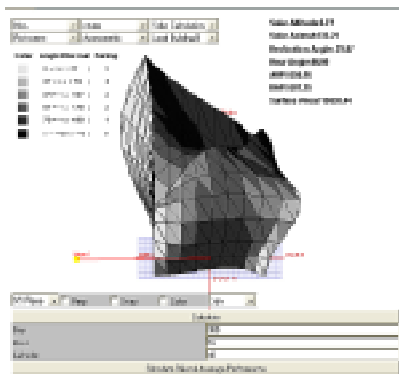


Fig 14 - Building B. Lat 40 N, Dec 21 at 16 Hours.

## Conclusion

Computer programs for architectural design have evolved to become very sophisticated and precise, but in the process require very steep learning curves to use them, and large amounts of data and time to produce useful results. Furthermore, most of them could be better integrated into the different areas of the design process. Very Simple Design Tools can be developed for use in the very first phases of the design process, using a minimum amount of data and with a strong influence on the design concept. EvSurf is a very easy to use program that permits the designed to analyse and compare complex building envelopes, with many surfaces as a function of direct solar radiation.

## References

- Rittel, Horst (1970). "Some Principles for the Design of an Educational System for Design", Design Methods and Theories, Vol. 20, N1, p 359-375.
- Olgyay, Aladar; Olgyay, Victor (1957). "Solar Control and Shading Devices". Princeton University Press, (reprinted 1976) Princeton NJ.
- La Roche, Pablo; Quiros, Carlos; Bravo, Gaudy; Gonzalez, Eduardo; Machado, Maria (2001). "Keeping Cool: Principles to avoid Overheating in Buildings" PLEA Notes, Design Tools and Techniques. Australia.
- Szokolay, Steven (1996). "Solar Geometry" PLEA Notes, Design Tools and Techniques. Australia.
- Han, Zhonglin; Milne Murray(2001). "Solar-2. A Computer Aided Design Tool". Department of Architecture and Urban Planning. University of California Los Angeles.
- Setiadarma, Effendi and Schiler, Marc (1996). "Shading Mask: A Computer-Based Teaching Tool for Sun Shading Devices." Proceedings of the American Solar Energy Conference, Asheville, NC.
- Bryan, Harvey; Autif, S. (2001). "Shading Analysis: Analog versus Digital Tools." April 21-25. Proceedings of the American Solar Energy Conference, Washington DC,.
- Milne, Murray et al (2001). "A Drag and Drop Energy Design Tool." Proceedings of the American Solar Energy Conference. June. Washington DC.
- Yezioro A.; Shaviv, Edna (1994). "A Design Tool for Analyzing Mutual Shading Between Buildings" Solar Energy. Vol. 52, N 1, (pp 27-37). Pergamon Press Ltd., USA
- Knowles, Ralph (2000). "The Solar Envelope, Its meaning for Urban Growth and Form". July. Architecture, City and the Environment, Proceedings of PLEA 2002. Cambridge UK.
- Schiler, M. & Ueng Fang, P. (1993). "Solvelope: An Interactive Computer Program for Defining and Drawing Solar Envelopes" 18th National Passive Solar Conference – ASES. Washington DC
- Gronbeck, Christopher (2000). <http://www.susdesign.com>
- Milne, Murray et al (2001). Department of Architecture and Urban Design, University of California Los Angeles. <http://www.aud.ucla.edu/energy-design-tools>
- Schiler, Mark (1999). University of Southern California; <http://www.usc.edu/dept/architecture/mbs/tools/index.html>
- Schiler, Marc (2000). Kensek, Karen and Noble, Doug. "Websites as a Computer Toolshed and Library: an Introduction to USC-MBS." June.
- La Roche, Pablo; Liggett, Robin (2001). "Very Simple Design Tools: A web Based Assistant for the Design of Climate Responsive Buildings" December. Architectural Science Review. Volume 44, N 4. p 437-448.
- Terzidis, Kostas (2002). "2D and 3D Environments in Java", Prentice Hall.
- Dildine, James P. (1999). "Triangle area calculator" [http://www.mste.uiuc.edu/dildine/tcd\\_files/program17.htm](http://www.mste.uiuc.edu/dildine/tcd_files/program17.htm)