## Surface Explorer 3D v3.0 (SurfX3D) Overview

Surface Explorer 3D plots mathematical surfaces in 3D. User defined surface equations can be explicit, implicit, or parametric. Surface and color equations are specified by user. The user can export to the clipboard, or file formats including VRML, POV, DXF, STL, and output to a printer.
Surface Explorer 3D is provided as freeware in the hopes that others can enjoy exploring math equations. Full MFC C++ source code is included as freeware.

Shown below is the main screen while plotting the Trefoil surface.
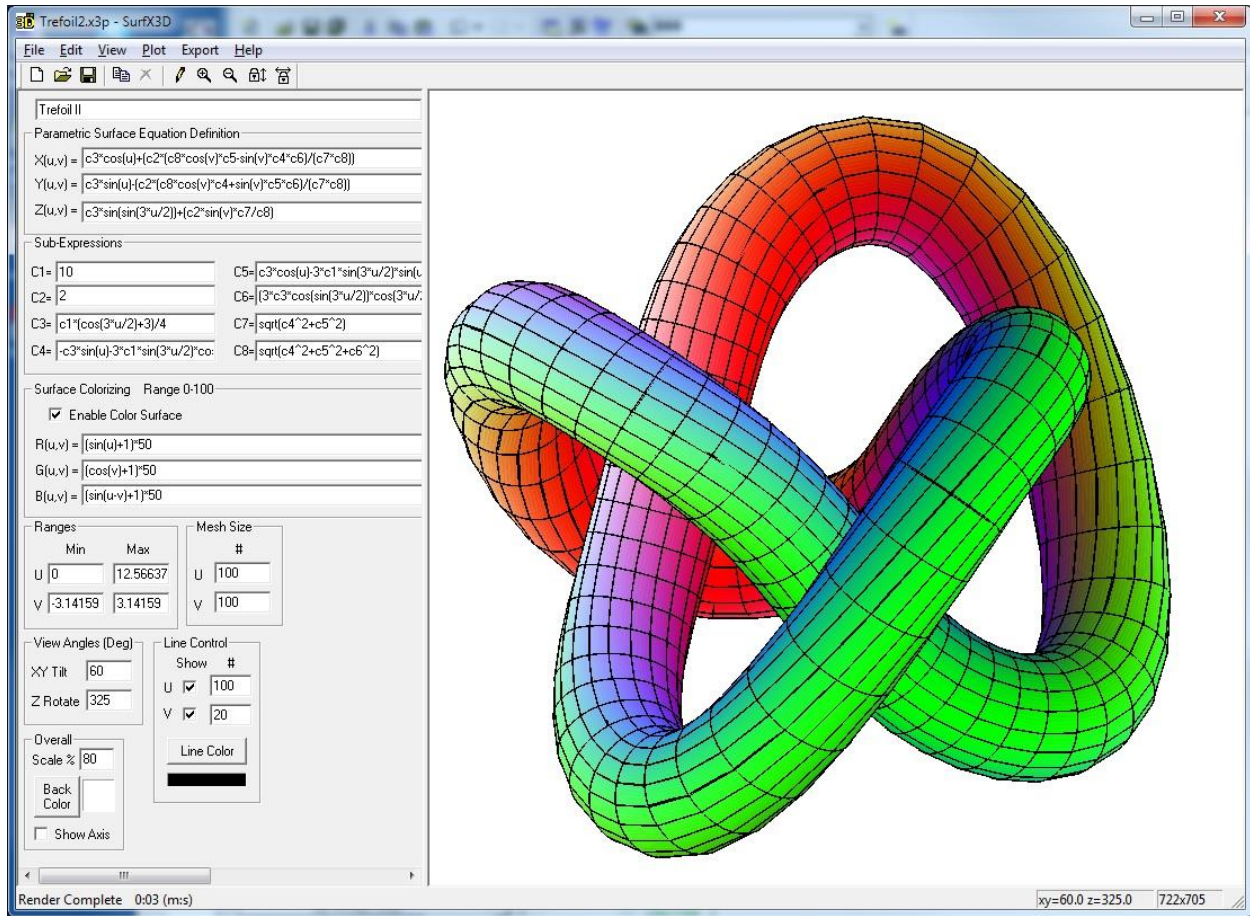
Surface Explorer 3D allows defining mathematical surfaces and plotting those surface equations in 3D. The Mobius strip shown below is a surface included with the samples.
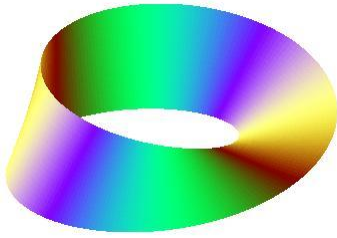
Three types of surface definitions are supported:

1) Parametric   X,Y,Z(u,v)
2) Explicit   Z=F(X,Y)
3) Implicit   F(X,Y,Z)=0

Surface equations are entered in a 'Basic-like' syntax such as:
        Z = sin ((X * Y) / 5)
Equations are case-insensitive (ie: upper or lower case does not matter).

A user definable colorizing function provides the ability to show the surface in varying degrees of color.

Various plot parameters can be specified:
- The viewing angles that the surface will be drawn from
- Viewable ranges of independent axis variables
- Number of contour lines to be drawn
- Drawing scale
- Hidden line removal
- Displaying coordinate axis
- Background color
- Line color

There are a number of built-in math functions including trigonometric functions, logarithm functions, absolute values, and others that can be used to define surface equations and colorizing functions.

Built-in Math Functions

The following mathematical operators are available:

Operator	Purpose
( )	grouping
+	addition
-	subtraction
*	multiplication
/	division
^	power

The following built-in math functions are available: case-insensitive

Function Syntax	Purpose
min(v1,v2)	min of two values
max(v1,v2)	max of two values
abs(v)	absolute value
sqrt(v)	square root
exp(v)	e ^ v
ln(v)	natural logarithm
log10(v)	base 10 logarithm
ceil(v)	next higher integer
floor(v)	next lower integer
rand()	random number
hypot(x,y)	hypotenuse: sides x,y
sin(radians)	trig sine
cos(radians)	trig cosine
tan(radians)	trig tangent
asin(v)	arc sine
acos(v)	arc cosine
atan(v)	arc tangent
sinh(v)	hyperbolic sine
cosh(v)	hyperbolic cosine
tanh(v)	hyperbolic tangent
atan2(y,x)	arc tangent (y / x)
sign(x)	-1 if x < 0 else +1

Parametric Surfaces X,Y,Z(u,v)

A parametric surface is defined using two independent parameters u and v.
X, Y, and Z are then defined using u and v. Thus, the surface is defined as:

X = X(u,v)
Y = Y(u,v)
Z = Z(u,v)
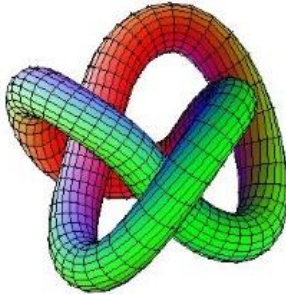
The color of the surface can be defined using u and v as follows:

R = R(u,v)   // red component   0-100
G = G(u,v)   // green component   0-100
B = B(u,v)   // blue component   0-100

Parametric Surface Plot Parameters X,Y,Z(u,v)

Surface Description
        Description of the surface

Parametric Surface Equation Definition:
        The equations of the surface can use:
            • u, v values
            • Built-in math functions
            • Sub-expressions (C1-C8)
        Provide the defining surface equations
                X(u,v) = [*X defining surface equation*]
                Y(u,v) = [*Y defining surface equation*]
                Z(u,v) = [Z *defining surface equation*]

Sub-Expressions (C1 – C8)
        Sub-Expressions can be used for defining parametric constants,
        repeated expressions, or just simplifying equations.
        They can use:
            • u, v values
            • Built-in math expressions.
            • Higher number expressions can use lower number expressions
              ie C6 can use C5,C4,C3,C2,C1
                    Example: C2=C1*sin(u)*cos(v)   is valid
                                    C3=3*C4   not valid

Surface Colorizing:

Enable Color Surface - must be checked to perform surface colorization.

The equations of the surface color can use:
- u, v values
- Built-in math functions
- Sub-expressions (C1-C8)

Note that these functions must produce a value between 0=off and 100=full on

Each video pixel consists of a red, green, and blue component.

R(u,v) = [*Red   defining surface color equation*]
G(u,v) = [*Green defining surface color equation*]
B(u,v) = [*Blue   defining surface color equation*]

Ranges

Specify the minimum and maximum values for u and v
for the viewable region of the surface.
Min value must be less than max value.
Also specify the number of u and v contour lines to draw.

Mesh Size

U,V - specify the number of intervals in each independent variable for surface resolution
Note: Larger values will take more rendering time (50-100 typical).

Viewing Angles (Degrees)

XY Tilt - angle of x-y plane and a plane perpendicular to line of view
A value of zero is looking at the surface as a top view. (typical 60)
Z Rotate - rotation angle about the z-axis. (typical 50)

Overall:

Scale % - used to reduce the image size (100=full scale)
Back Color - background screen color (use color selection button)
Show Axis - if checked, will draw the X, Y, Z coordinate axis.

Line Control

Draw U Lines – if checked, will draw the u contour lines
Draw V Lines – if checked, will draw the v contour lines
Line Color - the color used for drawing contour lines (use color selection button)

Explicit Surfaces Z = F(x,y)

An explicit surface is one of the simplest ways of defining a surface.
The Z value is defined as a function of X and Y. Thus, the surface equation is:

$$Z = F(x,y)$$

The surface is restricted to having one Z value for a given X and Y combination.
Not every surface can be defined explicitly.
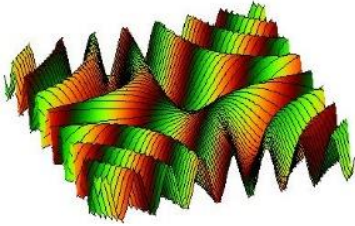
The color of the surface can be defined using x, y and z as follows:

```
R = R(x,y,z)     // red component     0-100
G = G(x,y,z)     // green component   0-100
B = B(x,y,z)     // blue component    0-100
```

Explicit Surface Plot Parameters   Z = F(x,y)

Surface Description
        Description of the surface

Explicit Surface Equation Definition:
        The equation of the surface can use:
- x, y values
- Built-in math functions
- Sub-expressions (C1-C8)

        Provide the defining surface equation
            Z(x,y) = [*Z defining equation*]

Sub-Expressions (C1 – C8)
        Sub-Expressions can be used for defining parametric constants,
        repeated expressions, or just simplifying equations.
        They can use:
- x, y values
- Built-in math expressions.
- Higher number expressions can use lower number expressions
ie C6 can use C5,C4,C3,C2,C1
           Example: C2=C1*sin(x)*cos(y)   is valid
                        C3=3*C4   not valid

Surface Colorizing:

Enable Color Surface - must be checked to perform surface colorization.

The equations of the surface color can use:

- x, y, z values
- Built-in math functions
- Sub-expressions (C1-C8)

Note that these functions must produce a value between 0=off and 100=full on

Each video pixel consists of a red, green, and blue component.

$R(x,y,z) = [$ *Red   defining surface color equation* $]$
$G(x,y,z) = [$ *Green defining surface color equation* $]$
$B(x,y,z) = [$ *Blue   defining surface color equation* $]$

Ranges

Specify the minimum and maximum values for X,Y,Z
  for the viewable region of the surface.

Min value must be less than max value.

Limit Z – if checked, will limit z-values to specified

Mesh Size

X,Y - specify the number of intervals in each axis for surface resolution

Note: Larger values will take more rendering time (50-100 typical).

Viewing Angles (Degrees)

XY Tilt - angle of x-y plane and a plane perpendicular to line of view

A value of zero is looking at the surface as a top view. (typical 60)

Z Rotate - rotation angle about the z-axis. (typical 50)

Overall:

Scale % - used to reduce the image size (100=full scale)

Back Color - background screen color (use color selection button)

Show Axis - if checked, will draw the X, Y, Z coordinate axis.

Line Control:

X Lines – checkmark to draw constant X lines, specify number of lines

Y Lines – checkmark to draw constant Y lines, specify number of lines

Line Color - the color used for drawing contour lines (use color selection button)

Implicit Surfaces   F(x,y,z) = 0

An implicit surface is expressed as a function of X, Y, Z equal to zero.
Thus, the equation of an implicit surface is:

$$F(x,y,z) = 0$$

The plotting algorithm uses an error function F(x,y,z) = Err.
In order for the plotting algorithm to work, Err must produce a sign change.
Otherwise, the root searching algorithm will not find a zero root.
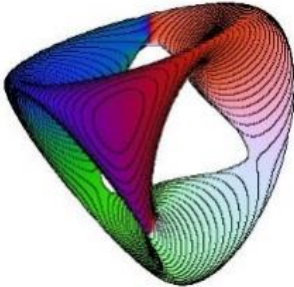
The color of the surface can be defined using x, y, and z as follows:

```
R = R(x,y,z)    // red component     0-100
G = G(x,y,z)    // green component   0-100
B = B(x,y,z)    // blue component    0-100
```

Suggestion:
   The plotting can be much slower than Parametric or Explicit surfaces.
   Reduce the size of the plot window to speed up drawing time.
   Once you are satisfied with the plot parameters, increase the
   drawing window size, redraw the plot and wait, wait, wait.

Implicit Surface Plot Parameters   F(x,y,z) = 0

Surface Description - Description of the surface

Implicit Surface Equation Definition:
        The equation of the surface can use:
            • x, y, z values
            • Built-in math functions
            • Sub-expressions (C1-C8)
        Provide the defining surface equation
                F(x,y,z) = [*defining surface equation*]

Sub-Expressions (C1 – C8)

 Sub-Expressions can be used for defining parametric constants,
 repeated expressions, or just simplifying equations.
 They can use:
- x, y, z values
- Built-in math expressions.
- Higher number expressions can use lower number expressions
  ie C6 can use C5,C4,C3,C2,C1

   Example: $C2=C1*sin(x)*cos(y)$ is valid
       $C3=3*C4$ not valid

Surface Colorizing:

 Enable Color Surface - must be checked to perform surface colorization.
 The equations of the surface color can use:
- x, y, z values
- Built-in math functions
- Sub-expressions (C1-C8)

 Note that these functions must produce a value between 0=off and 100=fullon
 Each video pixel consists of a red, green, and blue component.

  $R(x,y,z)$ = [*Red defining surface color equation*]
  $G(x,y,z)$ = [*Green defining surface color equation*]
  $B(x,y,z)$ = [*Blue defining surface color equation*]

Ranges

 Viewing range can be specified in one of two ways
 Specify the minimum and maximum values for X,Y,Z
 for the viewable region of the surface.
 Min value must be less than max value.

Mesh Size

 X,Y,Z - specify the number of search intervals in each axis for surface resolution
  Note: Larger values will take more rendering time (50-100 typical).

Viewing Angles (Degrees)

 XY Tilt - angle of x-y plane and a plane perpendicular to line of view
  A value of zero is looking at the surface as a top view. (typical 60)
 Z Rotate - rotation angle about the z-axis. (typical 50)

Overall:

 Scale % - used to reduce the image size (100=full scale)
 Back Color - background screen color (use color selection button)
 Show Axis - if checked, will draw the X, Y, Z coordinate axis.

Line Control:

 X,Y,Z – if checked, will draw the specified section cut lines
 Line Color - the color used for drawing contour lines (use color selection button)

Why SurfX3D? by Clark Dailey

The origins of SurfX3D span over forty years.

In 1974, as an engineering student, I became interested in plotting mathematical surfaces in 3D. I was inspired by another student, Nick Mykris, who had posters of 3D surface plots in the math department.

The explicit surface Z=F(X,Y) program was originally written in FORTAN on punch-cards which ran on a CDC 3400 mainframe and output to a drum line-plotter. No color. The 'advanced' FORTRAN-66 compiler would abort the batch job on the first syntax error it found, and not display any additional errors that were hiding in the code, waiting to be found on the next run.

The implicit surface F(X,Y,Z)=0 program was written a couple of years later for the same mainframe. The complex root searching needed for section cuts required much run and plot time for some of the more complex surface equations.

The code has gone through several incarnations with increasing functionality:

- 1974 – CDC 3400 mainframe output to a drum plotter.
      Written in FORTRAN-66 on punch cards.

- 1979 – IBM / 370 mainframe FORTRAN program output to a Tektronix line vector monitor

- 1987 – Compaq x386 portable PC written in GWBASIC.
      Every line of code in Gee-Wiz-Basic had a line number.

- 1988 – Windows 3.1 written in C.
      First use of color.

- 1998 – Windows NT written in Visual C++ MFC consisted of three separate programs
      PLOT3DZ3, PLOT3DF3, PLOT3DP1 and was submitted to public domain

- 2006 – SurfX3D 1.0 C++ MFC submitted to public domain

- 2007 – SurfX3D 2.0 C++ MFC submitted to public domain. Used OpenGL to allow rotating.

- 2021 – SurfX3D 2.5 C++ MFC submitted to public domain with export to STL for 3D printing

There have been software packages, such as Mathematica and Maple, that can render surfaces in 3D, but not everyone has access to these premium packages.

With the advent of the internet, the ability to plot mathematical surfaces in 3D has increased dramatically. There are some very amazing (free) programs, such as 'Persistence-of-Vision', which allows rendering the surfaces using lighting and ray tracing. Highly recommended.

In addition, the Virtual Reality Modeling Language (VRML) allows interactive viewing of surfaces by rotating and moving with the mouse. There are web browser VRML plug-ins that can be downloaded free of charge.

The Drawing Exchange File (DXF) export function has allowed users to generate data for 3D printing.

Exporting to STL format provides convenient support for 3D printing.

File Exporting (Export Menu)

STL File Format

Use this option to export the surface to STL format.
Popular for 3D printing.

Stereolithography STL File:

- File extension *.stl
- Popular file format for 3D printing
- Free on-line viewers
  https://3dviewer.net/

DXF Drawing File Format

Use this option to export to the CAD Drawing Exchange File (DXF) format.

Drawing Exchange File:

- File extension *.dxf
- CAD data file format
- Free on-line viewers
  https://allinpdf.com/dwg-viewer
  https://gallery.proficad.com/tools/autocad-viewer.aspx

VRML File Format

Use this option to export the surface to Virtual Reality Modeling Language format.

The Virtual Reality Modeling Language File:

- File extension *.wrl
- Allows viewing and rotating a surface in real-time with the mouse
- Free viewers for download
  https://castle-engine.io/view3dscene.php
- Download VRML web browser plug-in for viewing WRL files
  http://www.parallelgraphics.com/products/cortona/

POV File Format

Use this option to export to Persistence-of-Vision rendering package.

Persistence-of-Vision (POV) File:

- File extension *.pov
- Adds shadowing via ray tracing
- Download POV ray tracing program to view POV files
  http://www.povray.org/

Internet Links

3D Surface Equations:
http://xahlee.info/surface/gallery.html
https://silviana.org/gallery/hauser/
https://homepage.univie.ac.at/herwig.hauser/bildergalerie/gallery.html
https://en.wikipedia.org/wiki/Implicit_surface
https://wims.unice.fr/gallery/