Three dimensional function

This is my first on PC program from year 1991. The following graphics took the Intel 80283 processor full two minutes to be completed. The same year, I have sent the program to the *Bajtek* IT magazine (I added an example of drawing of the Cardan coupling). It was received but not published. It was my reaction to inept attempts to create 3-Dim graphics in that magazine.
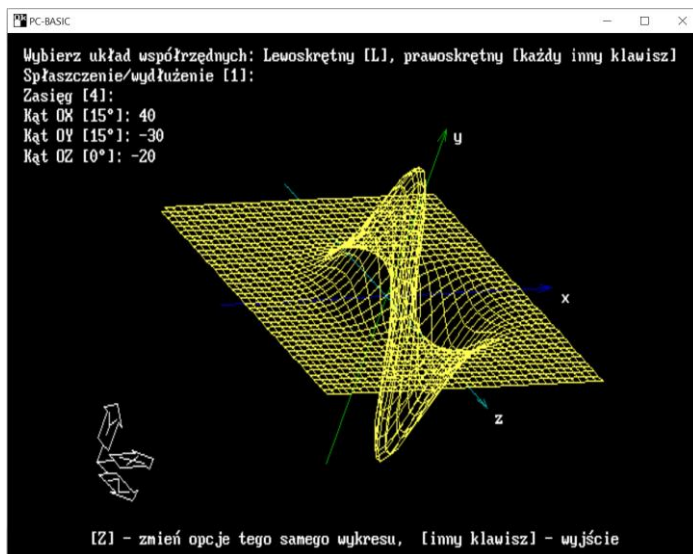
Below is a graphical image of the function:

```
V=-(X*X+Z*Z)
VV=-((X-2)*X+(Z-.5)*Z)
Y=5*EXP(V)-2*EXP(VV)
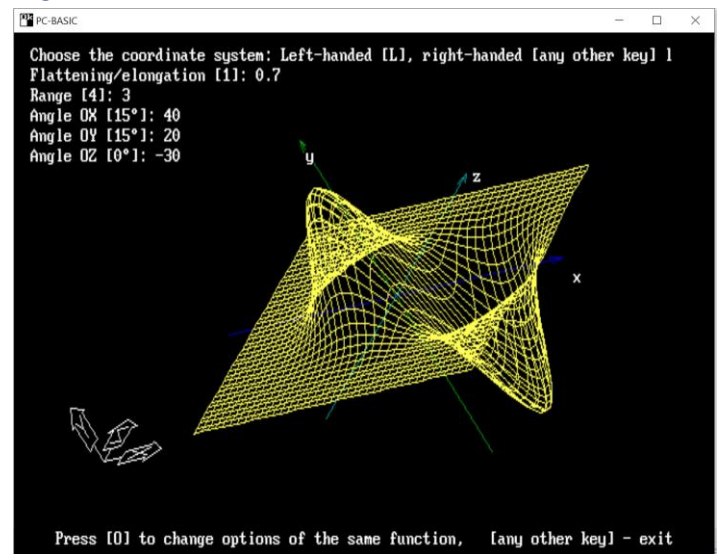```

that is mean:   $z = f(x,y) = 5*e^{-(x*x+y*y)} - 2*e^{-((x-2)*x+(y-.5)*y)}$

Polish version:                                              English version:



Unfortunately, the error handling is not refined in GWBasic. Below is providing two versions of the code and I am expecting the same result:

```
10        <-- beginning of a FOR loop below - two versions
20          PRINT "I=" I, "log(i)=" LOG(I);
30          ON ERROR GOTO 60
40          PRINT "         Nie ma bledu (No error)"
50          GOTO 90
60          A=ERR: B=ERL
70          PRINT: PRINT "Numer bledu (Error number)=" A "   Linia bledu (Error line)=" B
80          RESUME 90
90 NEXT
100 PRINT: PRINT "Proces zakonczony pomyslnie (Process completed successfully)"
110 END
```

First version: Second version:

```
10 FOR I=2 TO -2 STEP -1          <---->     10 FOR I=-2 TO 2 STEP 1
```

Result: Result:

```
I= 2         log(i)= .6931472   Nie ma bledu (No error)     I=-2          log(i)=
I= 1          log(i)= 0          Nie ma bledu (No error)    Illegal function call in 20
I= 0          log(i)=                                       OK
Numer bledu (Error number)= 5   Linia bledu (Error line)= 20
I=-1          log(i)=
Numer bledu (Error number)= 5   Linia bledu (Error line)= 20
I=-0          log(i)=
Numer bledu (Error number)= 5   Linia bledu (Error line)= 20
```

```
Process zakonczony pomyslnie (Process completed successfully)
OK
```

Very well - everything as expected   **Very bad - strage result !**

Error handling (ON ERROR GOTO...) doesn't work properly in GWBasic. Therefore, it is necessary to type just above the line of the function f(x,z), lines bypassing unworkable mathematical operations (colored lines below) by entering following condition prevents a dump:

      IF (*condition to bypass/unfavorable condition*) THEN BLAD=1: GOTO 14050

Example:

```
12090 IF (X+Z <= 0) THEN BLAD=1: GOTO 14050        ' this line bypasses the error
13000 Y=LOG(X+Z)                        ' <-- function itself y=f(x,z); there will be no error here
```

Another example:

```
12080 IF (X-2*Z <= 0) THEN BLAD=1: GOTO 14050        ' this line bypasses the error
12090 IF (X + LOG(X-2*Z) < 0) THEN BLAD=1: GOTO 14050 ' this line bypasses the error
13000 Y= 2 + SQR(X + LOG(X-2*Z))        ' <-- function itself y=f(x,z); there will be no error here
```

It is necessary to provide BLAD=1 (means in English: ERROR=1) informing the rest of the process that an incorrect mathematical operation has occurred.

I am sure that there is some methods of catching error 5 ('illegal function call'), but it must somehow artificially cheat the program. There is no such problem with error number 11 ('division by zero').
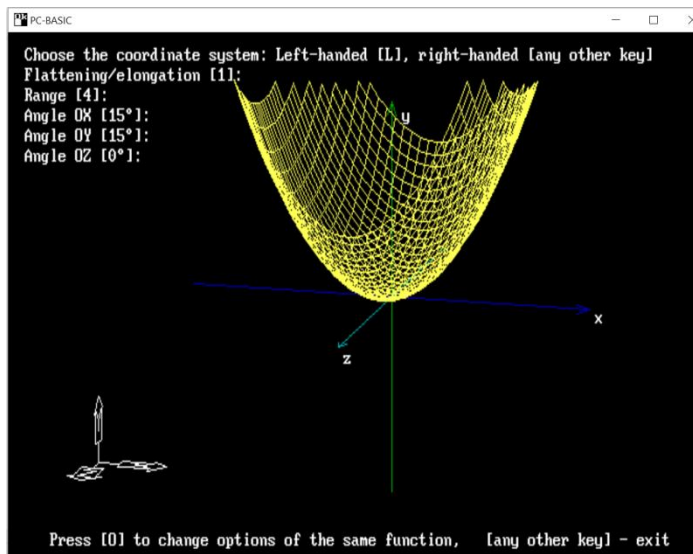
You have to be aware that any value outside the integer range (-32768 to 32767) causes an "OVERFLOW" error.

- - - - - - -

In the following examples I write X*X*X , although instead of it, you can always write X^3

13000 Y = X * X / 2 + Z * Z / 6
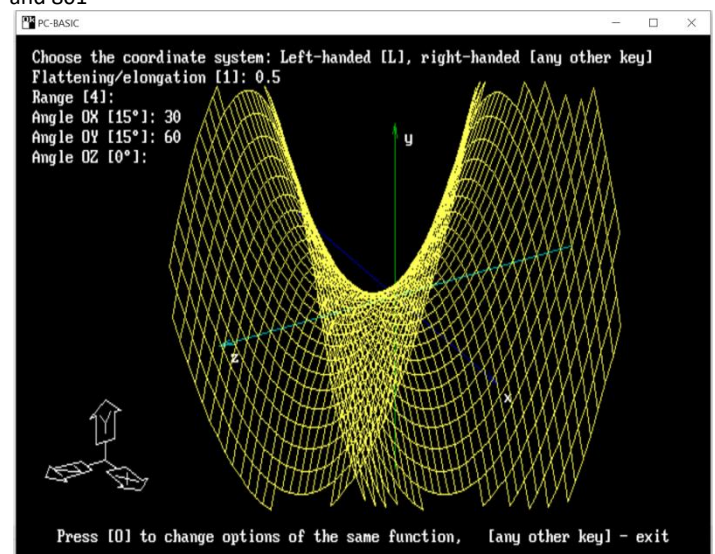$y = x^2/a^2 + z^2/b^2$ Elliptic paraboloid
pages 716 and 719, (2) Y=4x2+z2, pages 687 and 688, (4) page 756

13000 Y = Z * Z - X * X
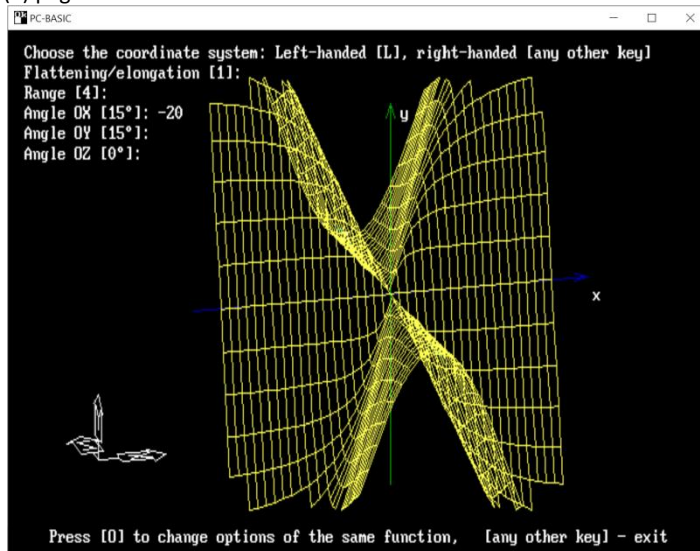$y = z^2/b^2 - x^2/a^2$ Hyperbolic paraboloid
 (1) pages 716 and 775, (2) pages 687 and 688 + 812, (4) pages 756 and 801
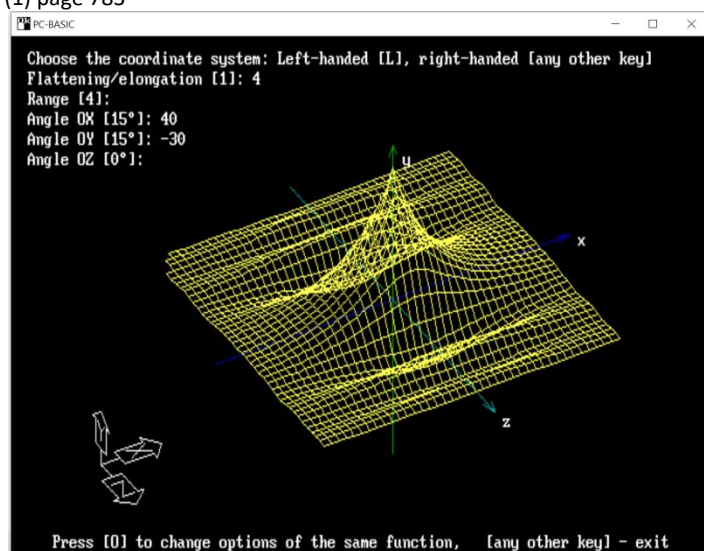
Y = (5*X*X*Z) / (X*X+Z*Z)

$y = 5x^2z/(x^2+z^2)$

(1) page 78

E = 2.71828   <-- line 194
13000 Y = COS(Z*Z)*E^(-SQR(X*X+Z*Z))
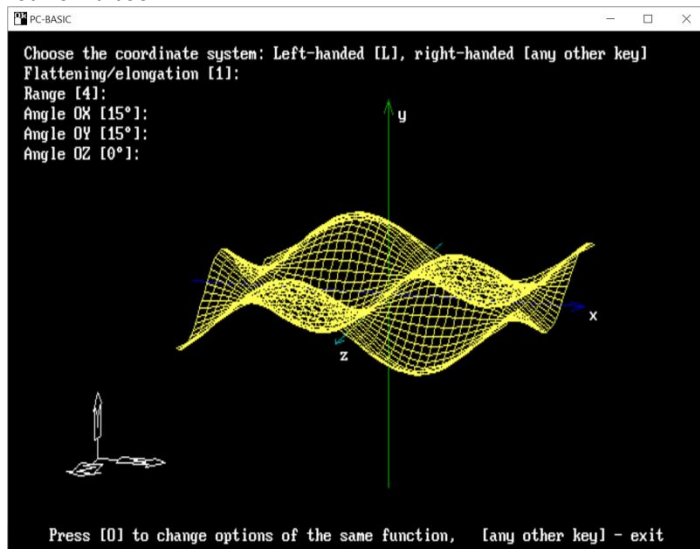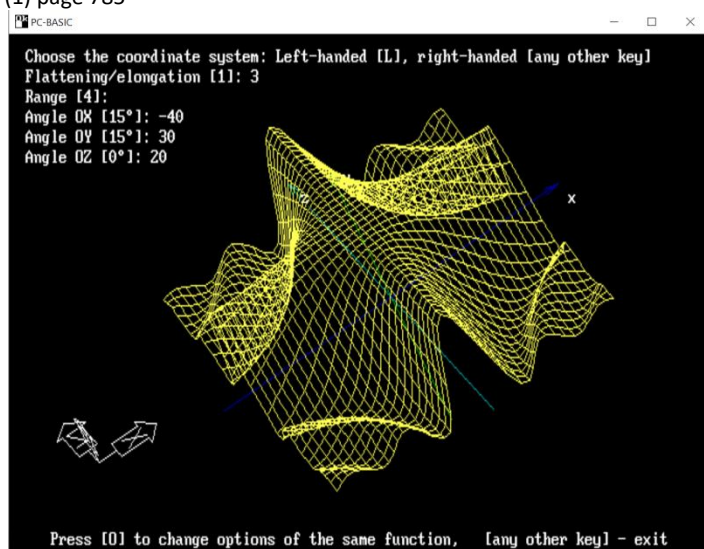
(1) page 783

13000 Y = SIN(X) * COS(Z)

y = sin x * cos z

not from a book

13000 Y = SIN(X*Z) / (X*Z)

(1) page 785

13000 Y = SIN(X * Z)

y = sin xz
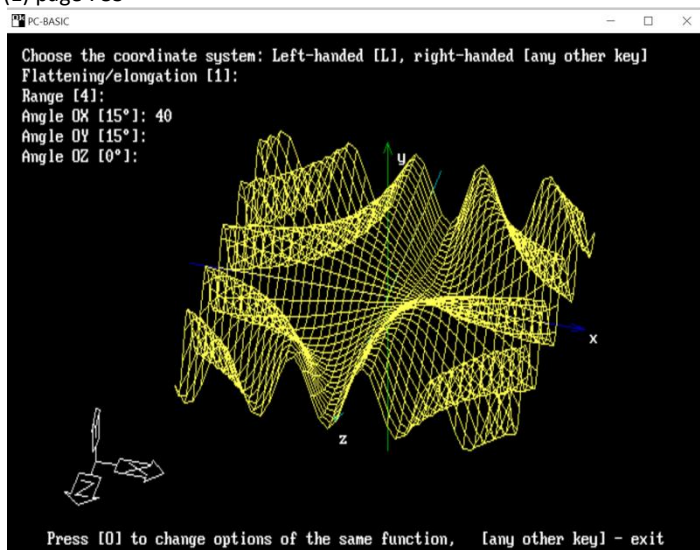
(1) page 783
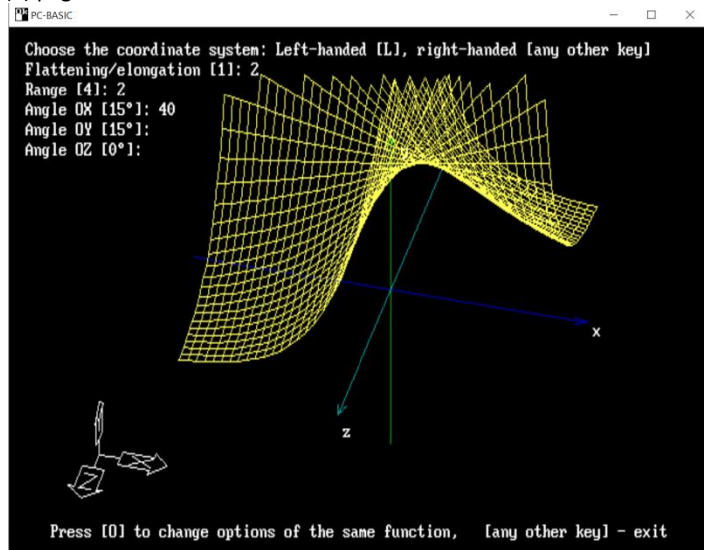

E = 2.71828   <-- line 194
13000 Y = E^(X*Z)

(1) page 786

13000 Y = SIN(X) + COS(Z)
(1) page 786
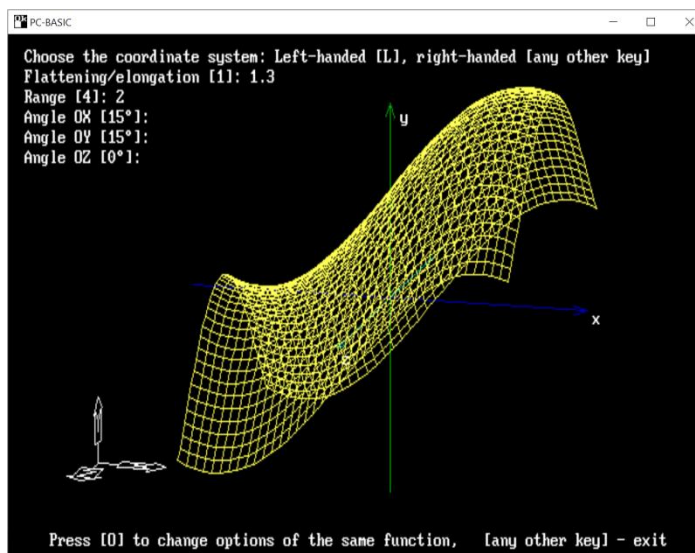
13000 Y = X*X*X -3*X*Z + Z*Z*Z
$y = x^3 - 3xz + z^3$
(1) page 836
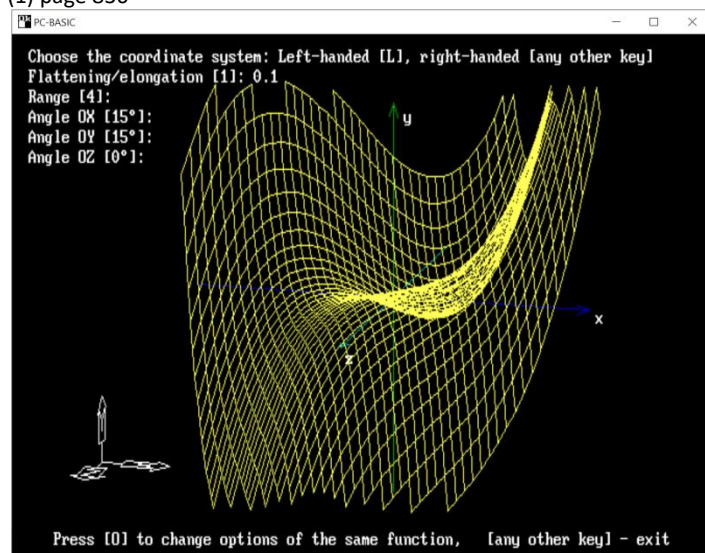
13000 Y = (X*Z) / (X*X+Z*Z)
$y = (xz)/(x^2+z^2)$
(1) page 786
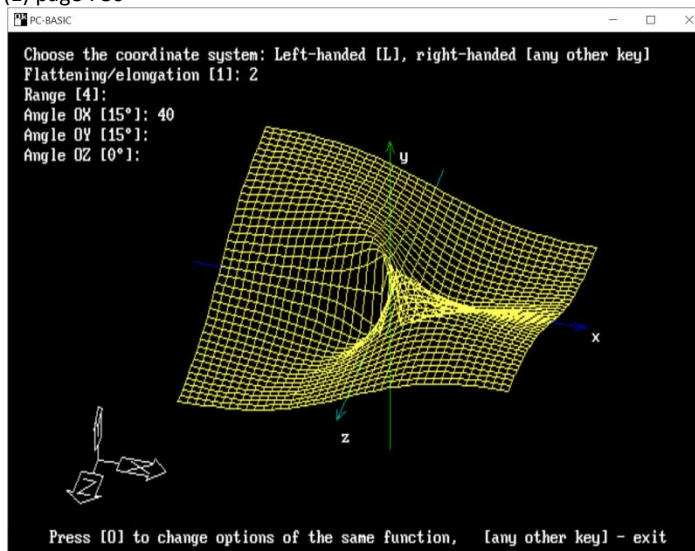
13000 Y = 4*X*Z-X*X*X*X-Z*Z*Z*Z
$y = 4xz-x^4-z^4$
(1) page 836
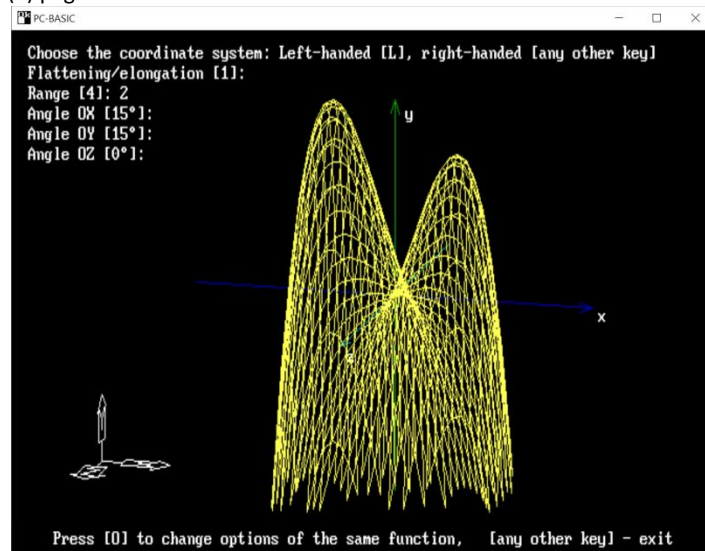
13000 Y = LOG(X*X+Z*Z)
$y = \log(x^2+z^2)$
(1) page 787
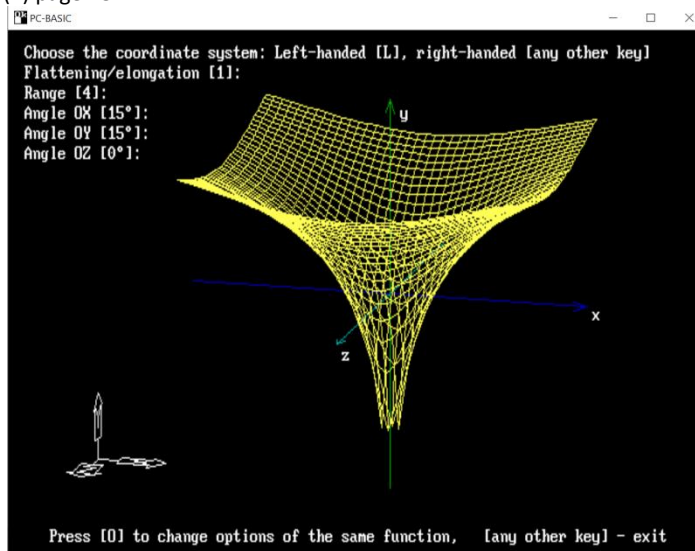
13000 Y = (3*X*X+1)/2 - X*(X*X+Z*Z)
$y = (3x^2+1)/2 - x(x^2+z^2)$
(1) page 836
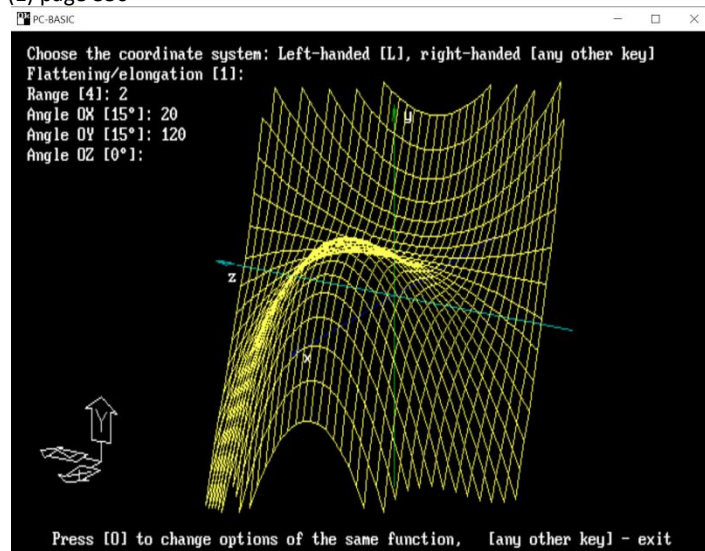
E = 2.71828   <-- line 194
13000 Y = (X*X+4*Z*Z)*E^(1-X*X-Z*Z)
y = $(x^2+4z^2)e^{1-xx-zz}$ (1) page 836
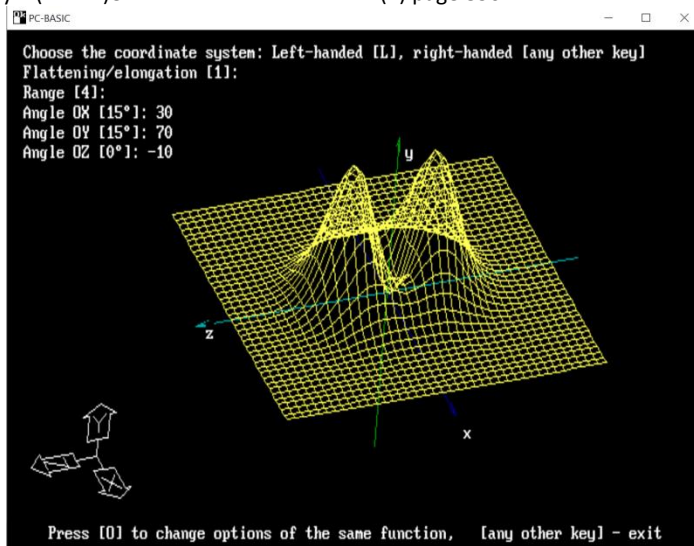
13000 Y = (-4*X)/(X*X+Z*Z+1)
y = $-4x/(x^2+z^2+1)$
(1) page 837
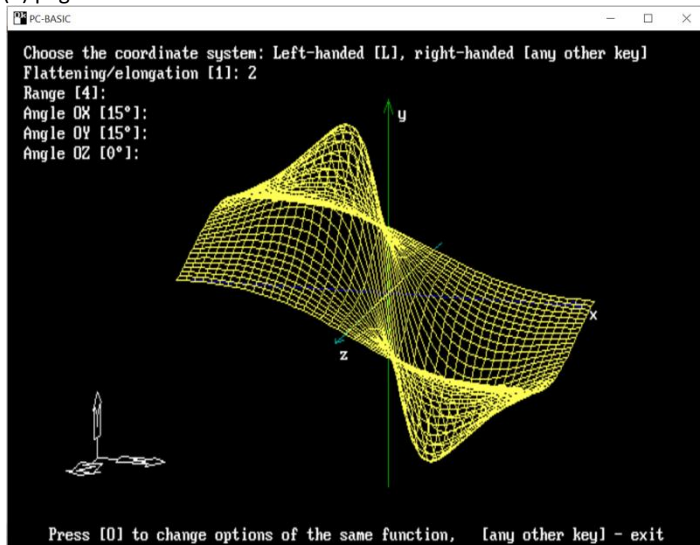
13000 Y = ATN(1/(X*X+Z*Z))
y = $\text{arctg}(1/(x^2+z^2))$ (1) page 837
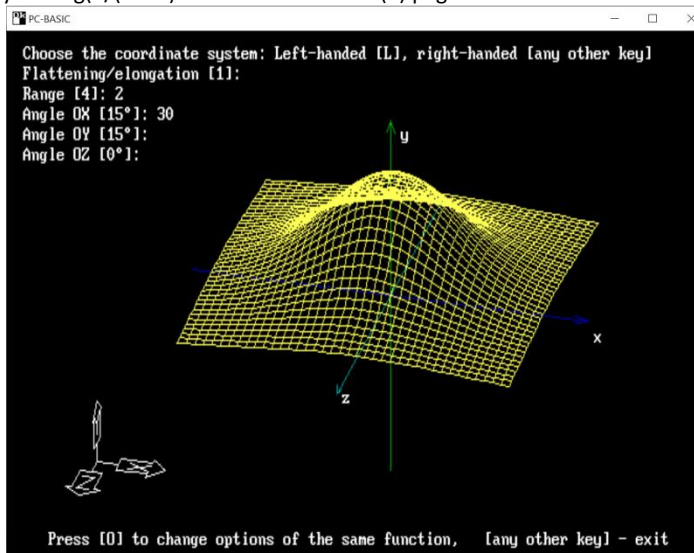
13000 Y = SIN(SQR(X*X+Z*Z))
(2) pages 758 and 759
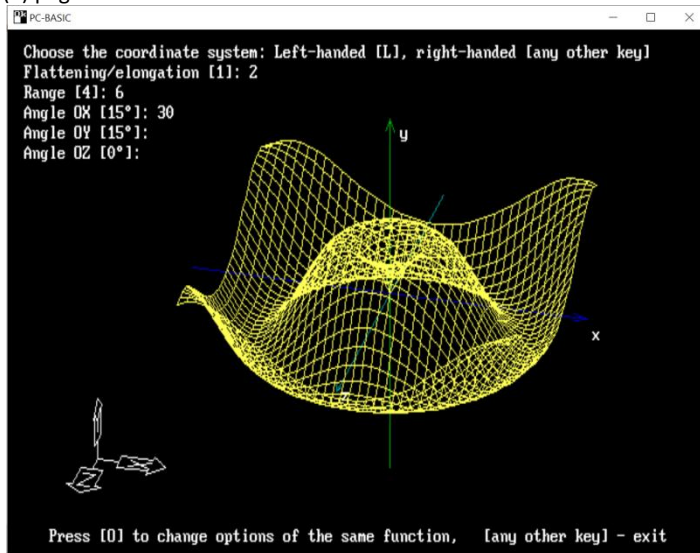
E = 2.71828   <-- line 194
13000 Y = E^(-(X*X+Z*Z))
y = $e^{-(xx+zz)}$
(1) page 837
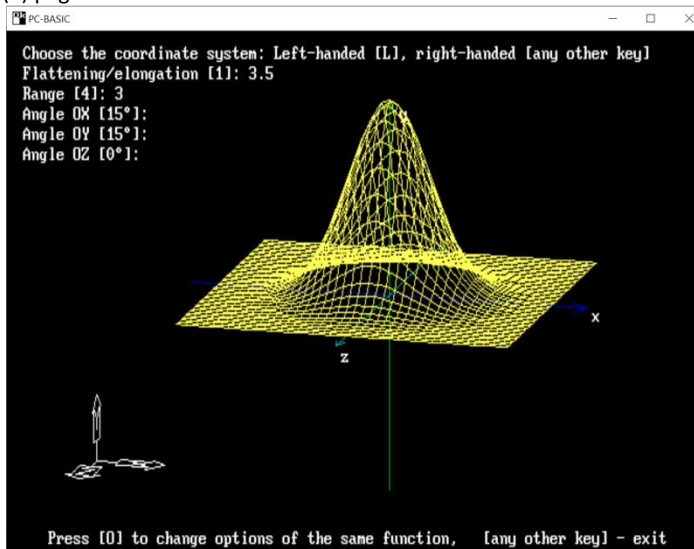
E = 2.71828   <-- line 194
13000 Y = X*X*Z*Z*E^(-X*X-Z*Z)
y = $x^2z^2e^{-xx-zz}$
(2) pages 758 and 759
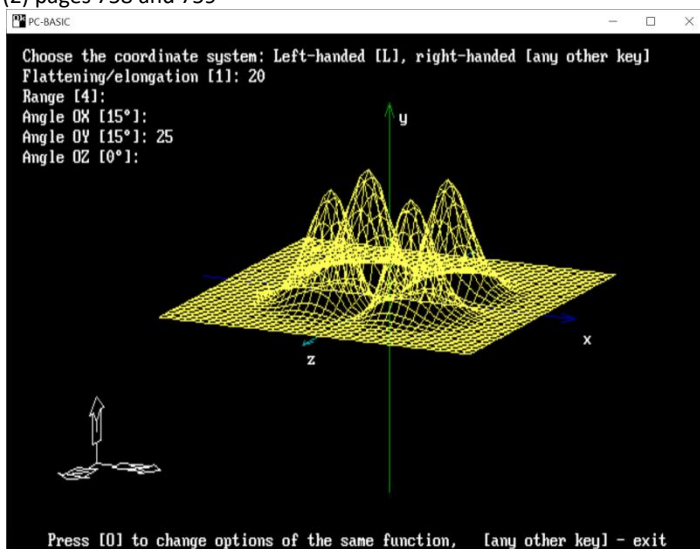
13000 Y = X*X*X-3*X*Z*Z
$y = x^3 - 3xz^2$
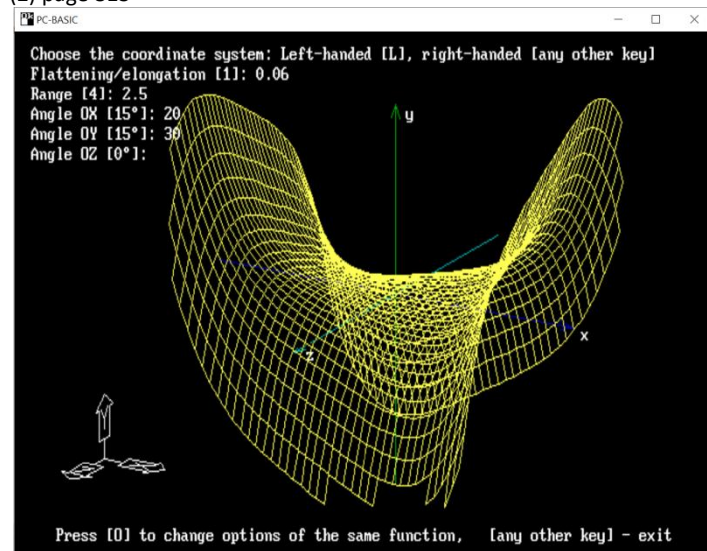(2) pages 758 and 759
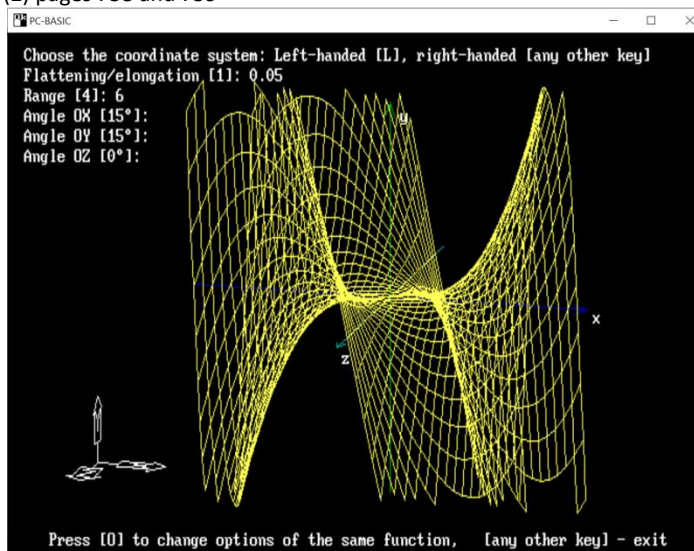
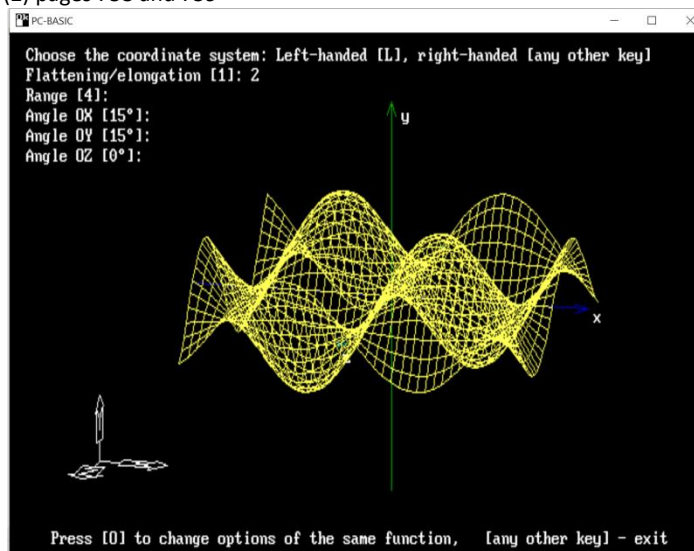## 13000 Y = X*X*X*X - Z*Z*Z*Z - 4*X*Z + 1
$y = x^4 - z^4 - 4xz + 1$
(2) page 813

13000 Y = SIN(X) * SIN(Z)
$y = \sin x * \sin z$
(2) pages 758 and 759

E = 2.71828   <-- line 194
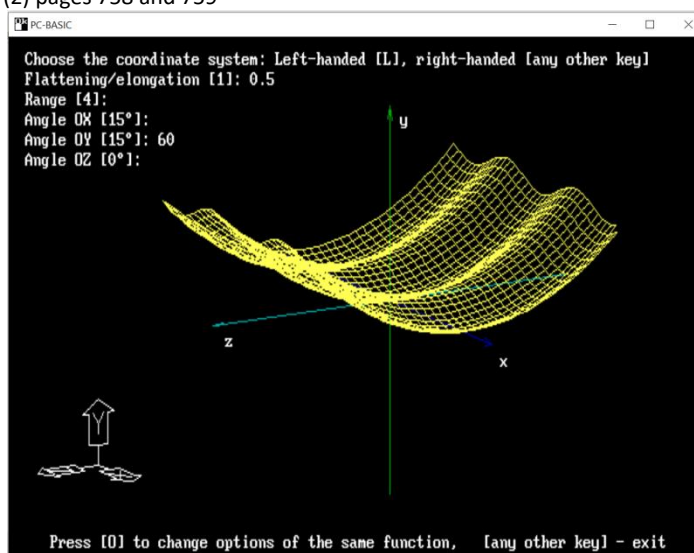## 13000 Y = -X*Z*E^(-X*X-Z*Z)
$y = -xze^{-x*x-z*z}$ (2) page 754
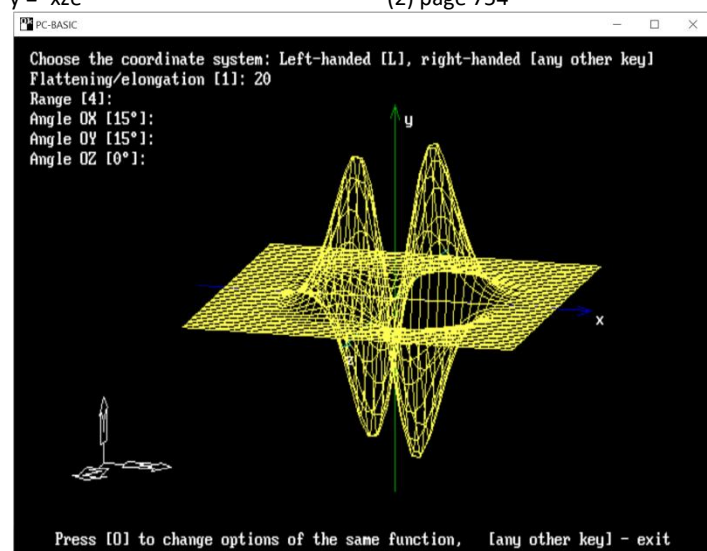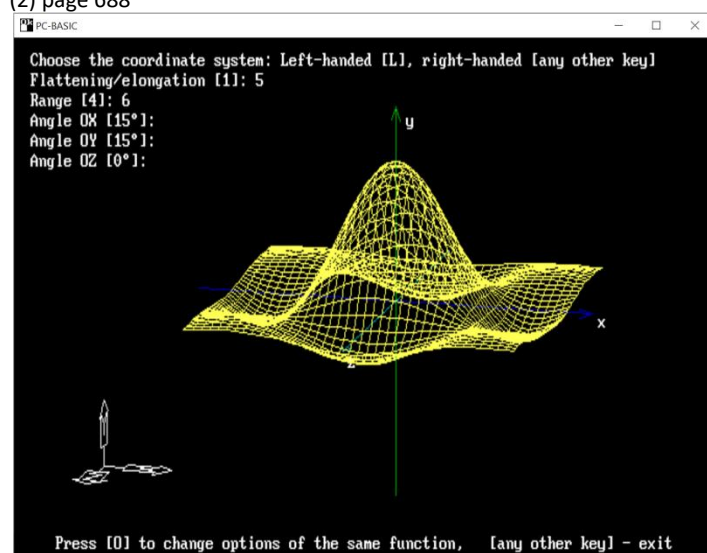

13000 Y = (SIN(X))^2 + 0.25*Z*Z
$y = \sin^2 x + 1/4\, z^2$
(2) pages 758 and 759

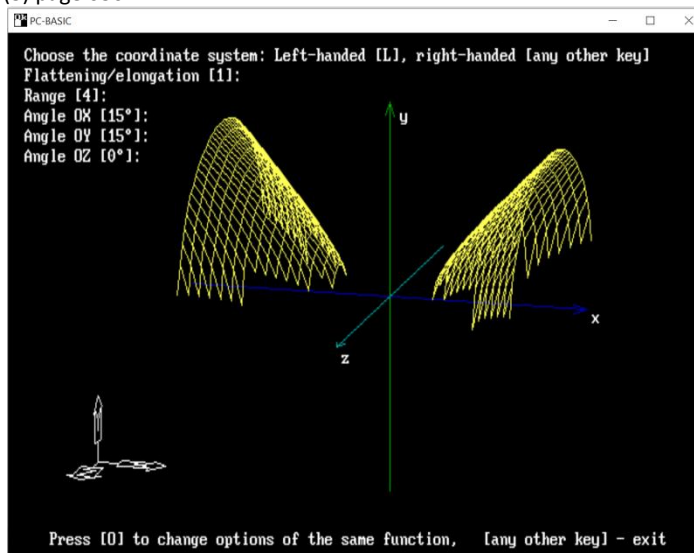## 13000 Y = (SIN(X)*SIN(Z)) / (X*Z)
$y = \sin x \sin z / (xz)$
(2) page 688

12990 IF X*X-Z*Z-1 < 0 THEN BLAD=1: GOTO 14050
13000 Y = SQR(X*X-Z*Z-1)
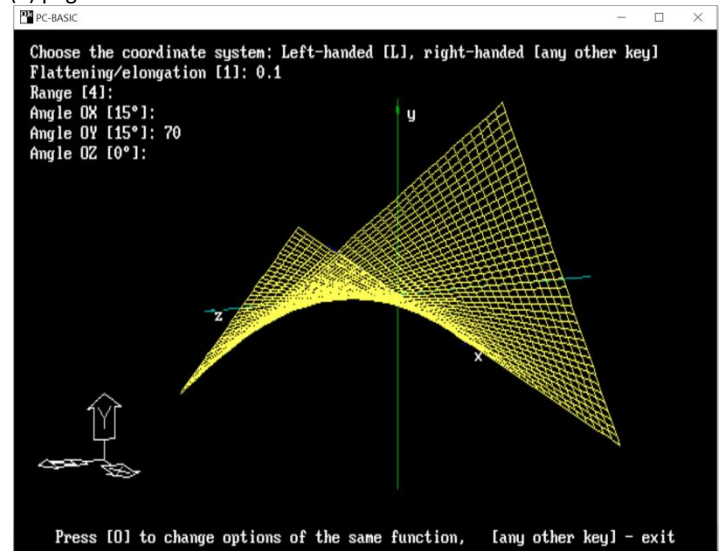(3) page 636

13000 Y = 2*X*Z   <-- saddle
y = 2xz
(3) page S-141

12990 IF X*X+Z*Z-1 < 0 THEN BLAD=1: GOTO 14050
13000 Y = SQR(X*X+Z*Z-1)
(3) page 635
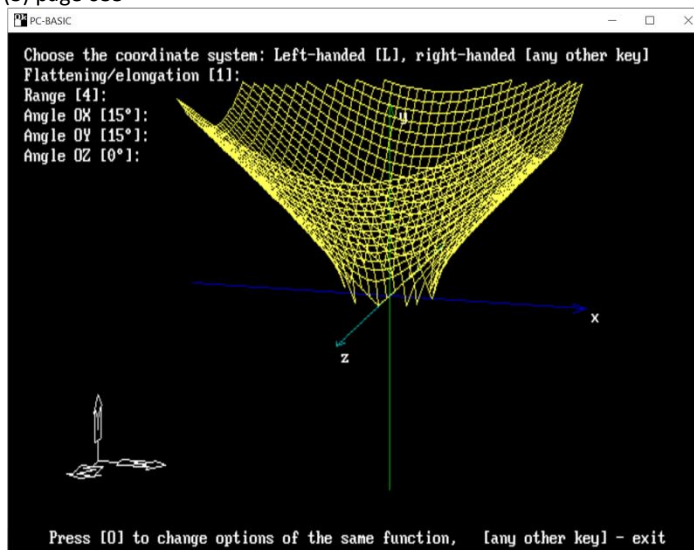
13000 Y = TAN(X*Z)
y = tan(xz)
(not from a book)

- - - - - - - - - - - - - - - - - -
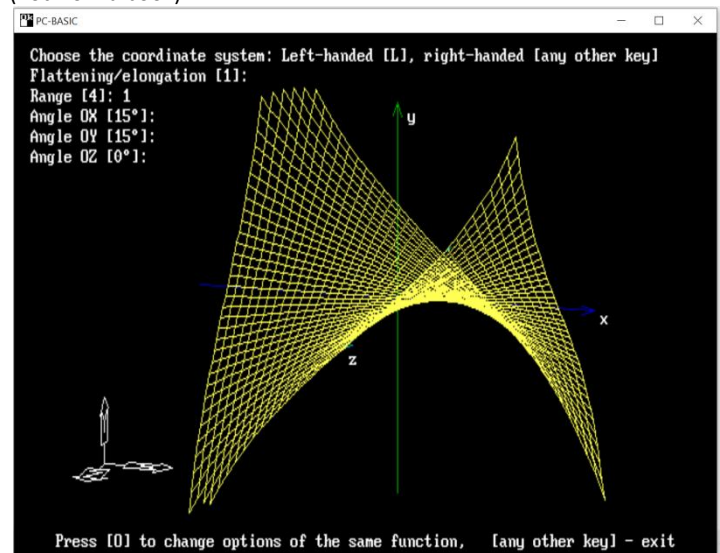
(1) Roland E. Larson, Robert P. Hostetler: Calculus with Analytic Geometry; The Pennsylvania State University, The Behrend College; Third edition; 1986, D.C. Heath and Company, ISBN: 0-669-09568-0
(2) James Stewart: Calculus, Concepts and Contexts; McMaster University; 1998, Brooks Cole Publishing Company, ISBN: 0-534-34330-9
(3) Sherman K. Stein: Calculus and Analytic Geometry; University of California, Davis; Fourth edition; 1987, McGraw-Hill Book Company, ISBN: 0-07-061159-9
(4) Harley Flanders: Calculus; Florida Atlantic University; 1985, W. H. Freeman and Company, ISBN: 0-7167-1643-7

Below is an example of using the program to make the generally understood spatial graphics.

Some translation:

Polish	English
Sprzeglo Oldhama	Oldham coupling
Pozycja pierwszego elementu (0 - 100)	Position of the first element (0 - 100)
Pozycja drugiego elementu (0 - 100)	Position of the second element (0 - 100)
Wybierz układ wspolrzednych: Lewoskretny [L] lub prawoskretny [inny klawisz]:	Select the coordinate system: Left-hand [L] or right-hand [other key]:
Czy chcesz widziec osie ukladu wspołrzednych (T - TAK; inny klawisz - NIE) ?	Do you want to see the axes of the coordinate system (T - YES; other key - NO) ?
Kat OX (°)	Angle OX (°)
Kat OY (°)	Angle OY (°)
Kat OZ (°)	Angle OZ (°)
Przycisnij dowolny klawisz aby wyjsc z programu	Press any key to exit the program
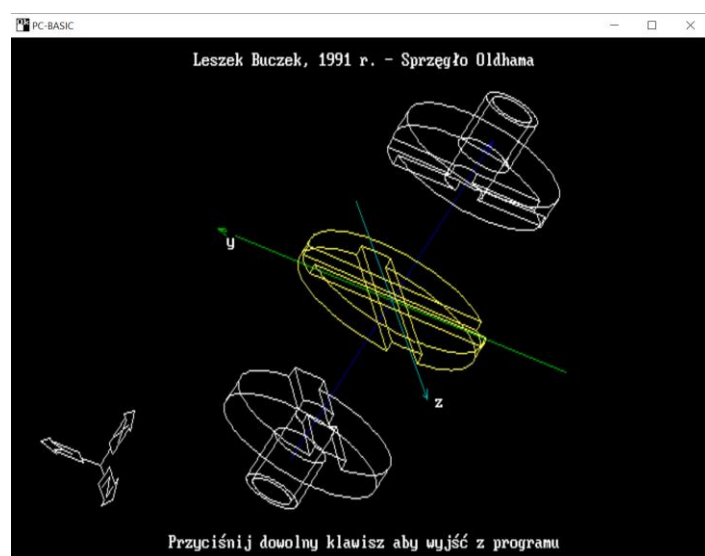
How other applications do it ?

The first function given above as an example.

' --- Function equation Y=f(X,Z) ---
V=-(X*X+Z*Z)
VV=-((X-2)*X+(Z-.5)*Z)
Y=5*EXP(V)-2*EXP(VV)
' -------------------------------

How *GeoGebra* does it ?

GeoGebra Wykres 3D ← https://www.geogebra.org/3d?lang=pl
$f(x,y) = 5*e^{-(x*x+y*y)}-2*e^{-((x-2)*x+(y-.5)*y)}$

GeoGebra Grafika 3D   ＜   ⚏   ZALOGUJ SIĘ...

⚪ $a(x, y) = 5\,e^{-(xx+yy)} - 2\,e^{-((x-2)x+(y-0.5)y)}$ ⋮   ↰

＋ Wprowadź...

△ GeoGebra Grafika 3D
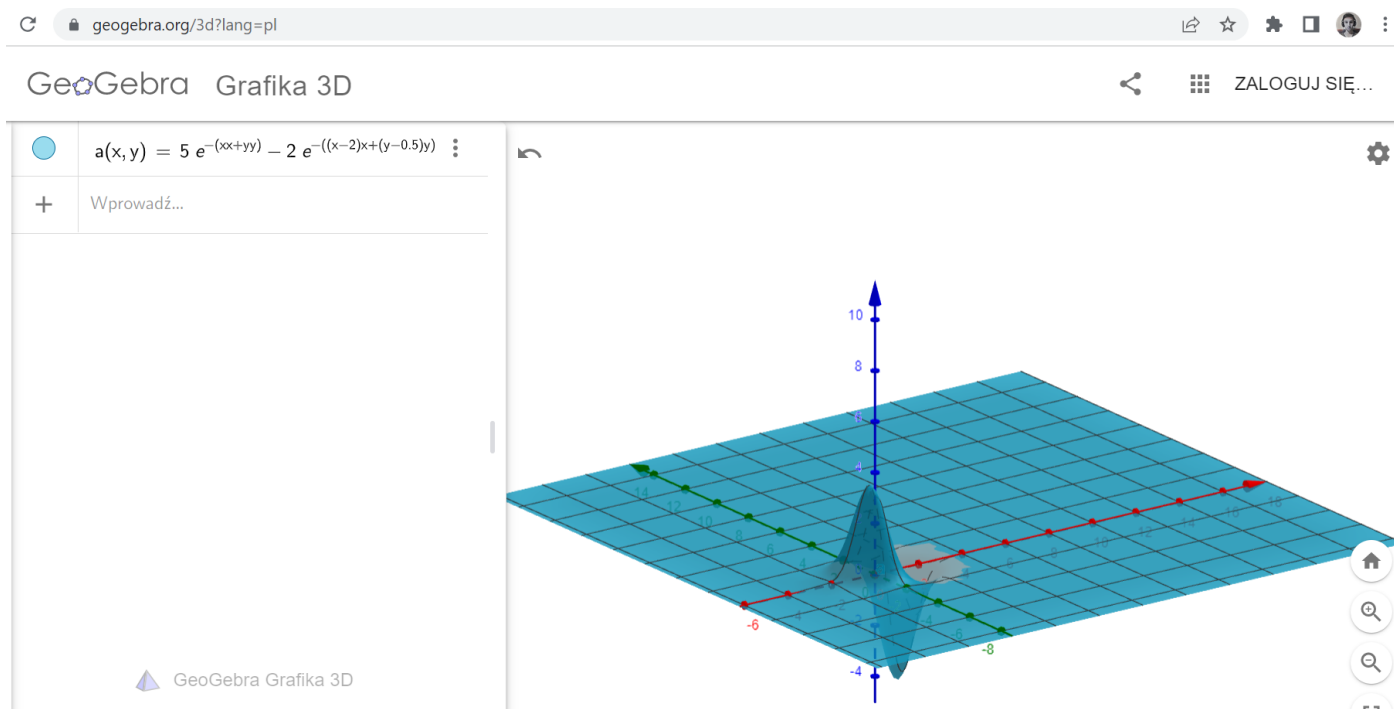
Generally correct, but apart from rotations I cannot do anything about it.

How *Python* does it ?

To obtain an image of a spatial figure in any of its settings, you need to import such <u>miraculous mechanisms</u> as:
- OpenGL.GL/GLU/GLUT
- vpython
or something similar.

If I give them up, what is left to me is the *matplotlib* (below is also *numpy*, which I do not need to use it here). The result is good, but only in one setting and position - a typically one-sided application.

Below are both the code to copy to Python and the result:

```
from mpl_toolkits import mplot3d
import numpy as np
import matplotlib.pyplot as plt

def f(x, y):
    return 5*np.exp(-(x*x+y*y)) - 2*np.exp(-((x-2)*x+(y-0.5)*y))
# V=-(X*X+Z*Z)
# VV=-((X-2)*X+(Z-.5)*Z)
# Y=5*EXP(V)-2*EXP(VV)

x = np.linspace(-3, 4, 20)
y = np.linspace(-3, 4, 20)

X, Y = np.meshgrid(x, y)
Z = f(X, Y)

fig = plt.figure(figsize=(12, 18))
ax = plt.axes(projection ='3d')
```
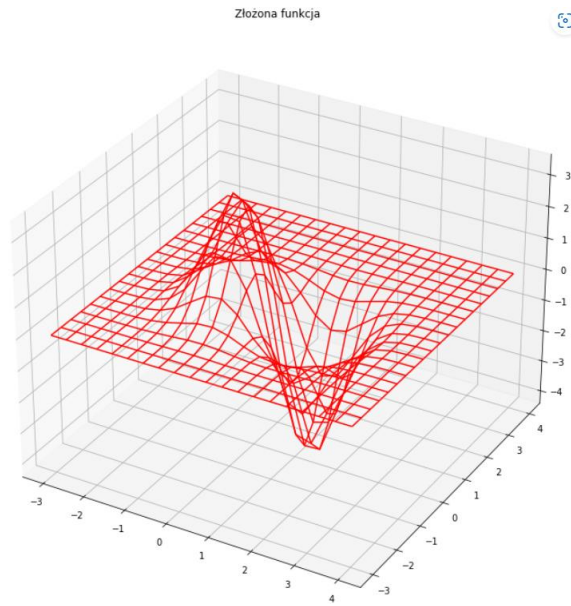
```
ax.plot_wireframe(X, Y, Z, color ='red')
ax.set_title('Złożona funkcja');
```


Złożona funkcja

Such things would probably do even Excel.