Three dimensional function

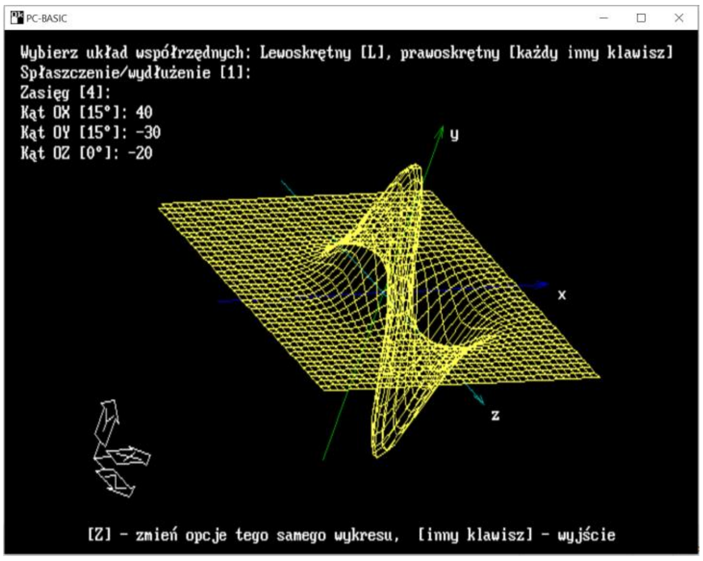
This is my first on PC program from year 1991. The following graphics took the Intel 80283 processor full two minutes to be completed. The same year, I have sent the program to the *Bajtek* IT magazine (I added an example of drawing of the Oldham coupling). It was received but not published. It was my 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)
```

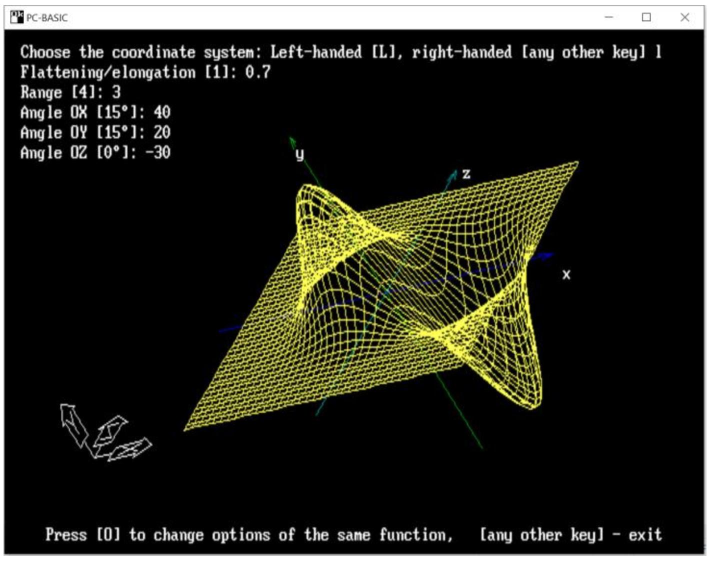
that is mean: $y = f(x,z) = 5*e^{-(x*x+z*z)}-2*e^{-((x-2)*x+(z-.5)*z)}$

Polish version:



```
Wybierz układ współrzędnych: Lewoskrętny [L], prawoskrętny [każdy inny klawisz]
Spłaszczenie/wydłużenie [1]:
Zasięg [4]:
Kąt OX [15°]: 40
Kąt OY [15°]: -30
Kąt OZ [0°]: -20

     [Z] - zmień opcje tego samego wykresu,  [inny klawisz] - wyjście
```

English version:



```
Choose the coordinate system: Left-handed [L], right-handed [any other key] l
Flattening/elongation [1]: 0.7
Range [4]: 3
Angle OX [15°]: 40
Angle OY [15°]: 20
Angle OZ [0°]: -30

  Press [O] to change options of the same function,   [any other key] - exit
```

Unfortunately, the error handling is not refined in GW-Basic. 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:

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

<---->

Second version:

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

Result:

```
I= 2          log(i)= .6931472   Nie ma bledu (No error)
I= 1          log(i)= 0          Nie ma bledu (No error)
I= 0          log(i)=
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)=
```

Result:

```
I=-2          log(i)=
Illegal function call in 20
OK
```

```
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
```

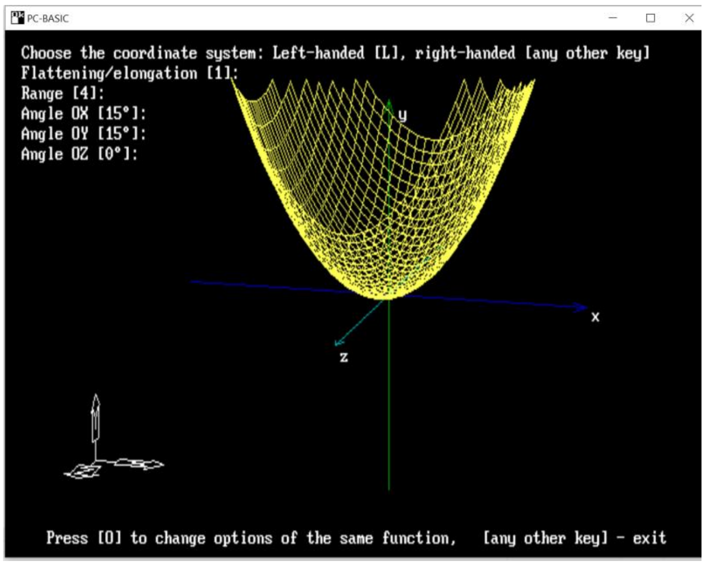
It is necessary to provide BLAD=1 (means in English: ERROR=1) informing the rest of the process that an incorrect mathematical operation has occurred.
I am sure that there is some methods of catching error 5 ('illegal function call'), but it must somehow artificially cheat the program. There is no such 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.

- - - - - - -

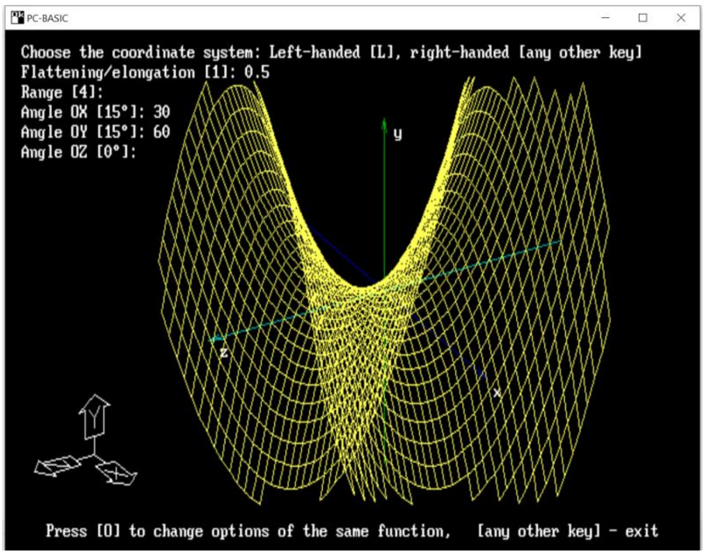
In the following examples I write X*X*X , although instead of it, you can always write X^3

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



```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

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

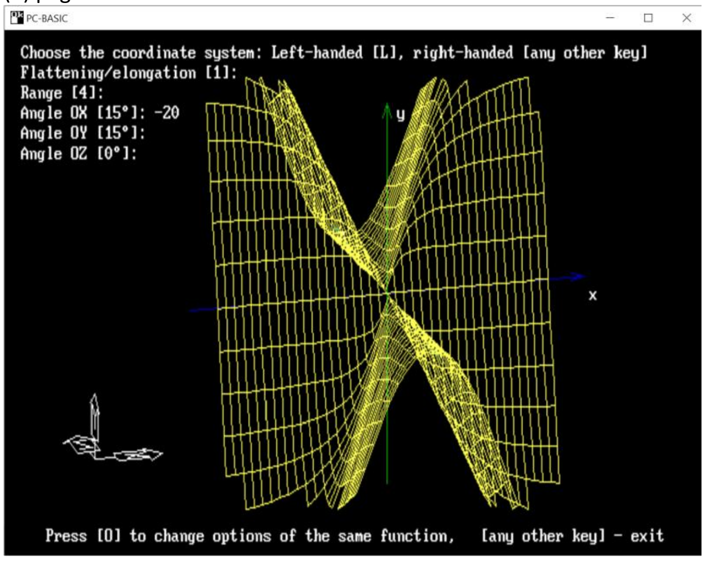


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.5
Range [4]:
Angle OX [15°]: 30
Angle OY [15°]: 60
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

13000 Y = (5*X*X*Z) / (X*X+Z*Z)

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

(1) page 78

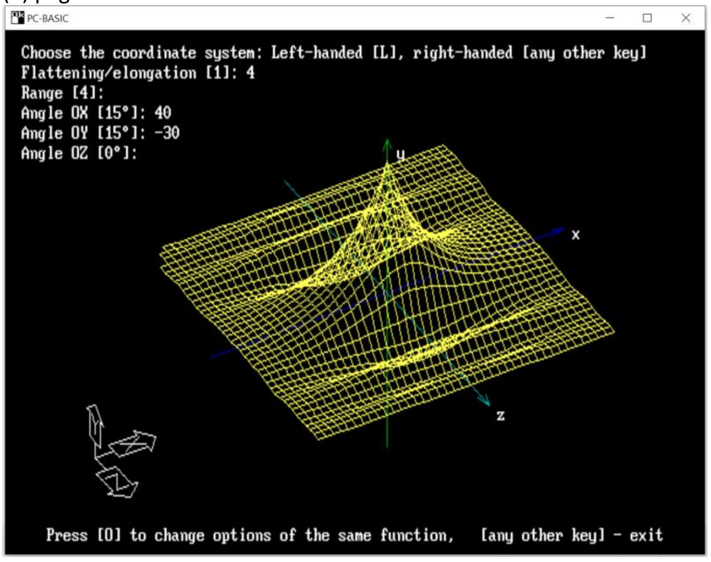


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]: -20
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = COS(Z*Z)*E^(-SQR(X*X+Z*Z))

(1) page 783

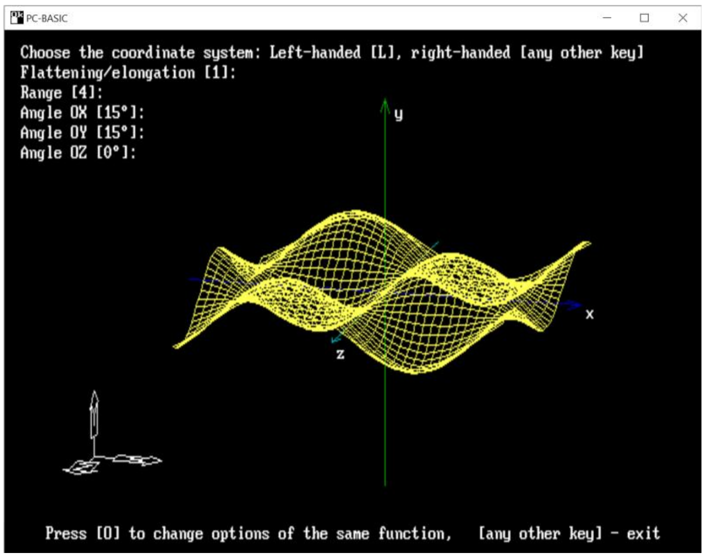


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 4
Range [4]:
Angle OX [15°]: 40
Angle OY [15°]: -30
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

y = sin x * cos z

not from a book

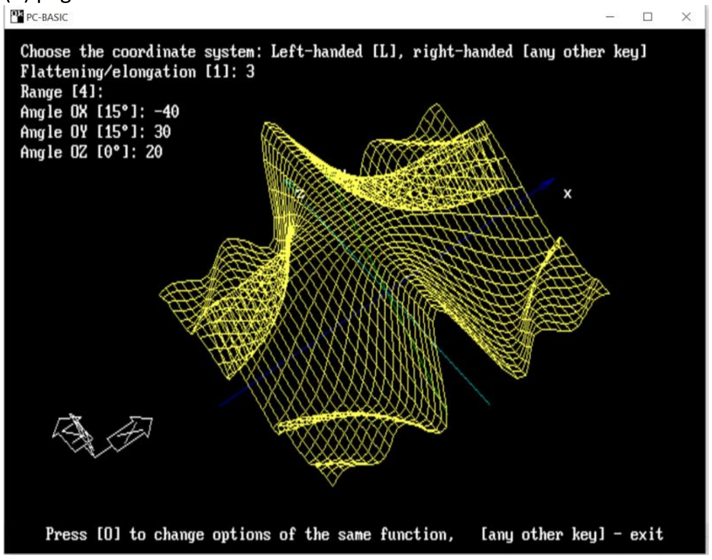


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

y = sin (xz) / (xz)

(1) page 785

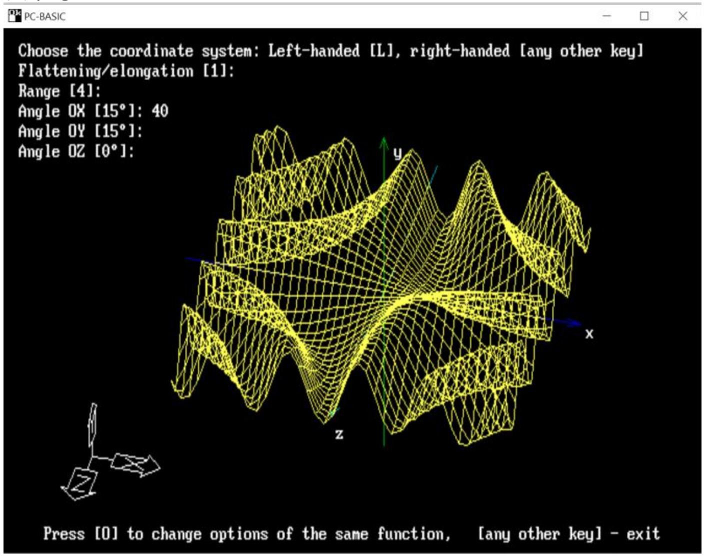


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 3
Range [4]:
Angle OX [15°]: -40
Angle OY [15°]: 30
Angle OZ [0°]: 20

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = SIN(X * Z)

y = sin xz

(1) page 783

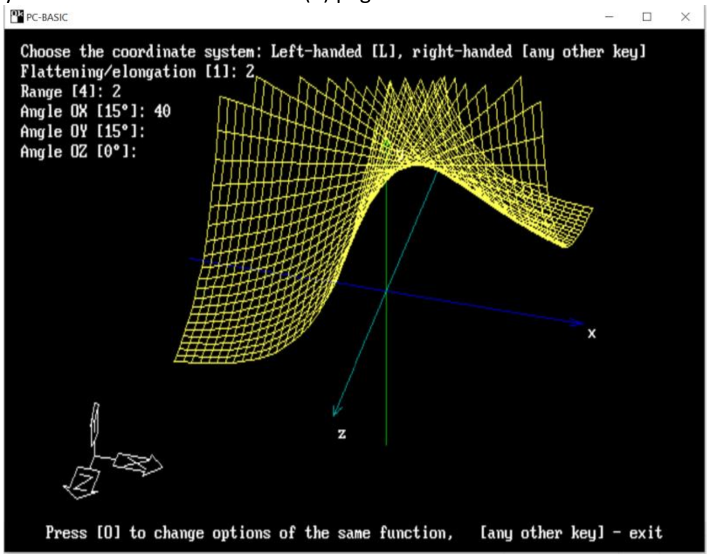


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]: 40
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = E^(X*Z)

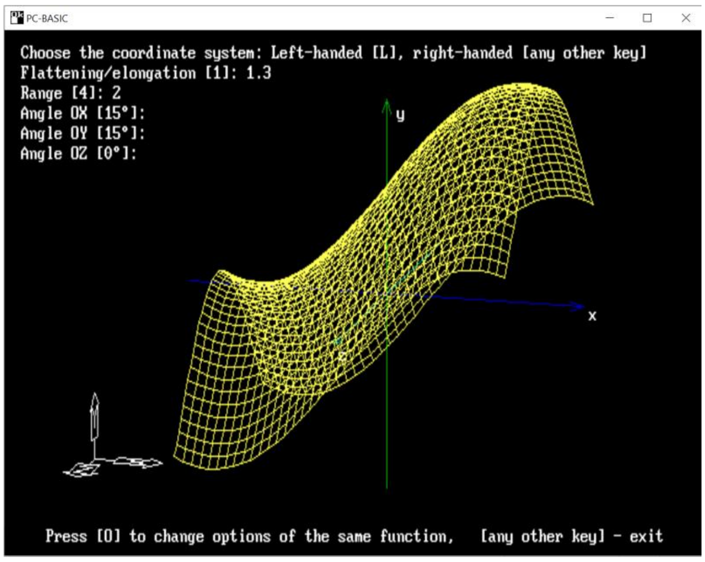
$y = e^{xz}$ (1) page 786



```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 2
Range [4]: 2
Angle OX [15°]: 40
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

(1) page 786



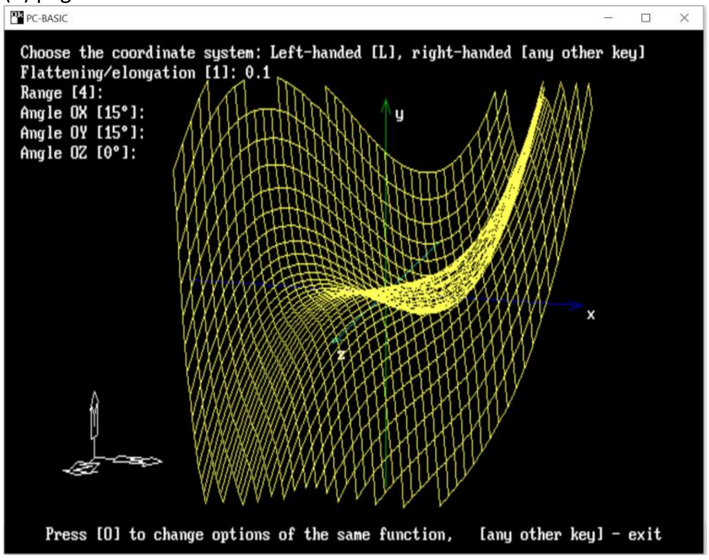
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 1.3
Range [4]: 2
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = X*X*X -3*X*Z + Z*Z*Z

y = x^3-3xz+z^3

(1) page 836



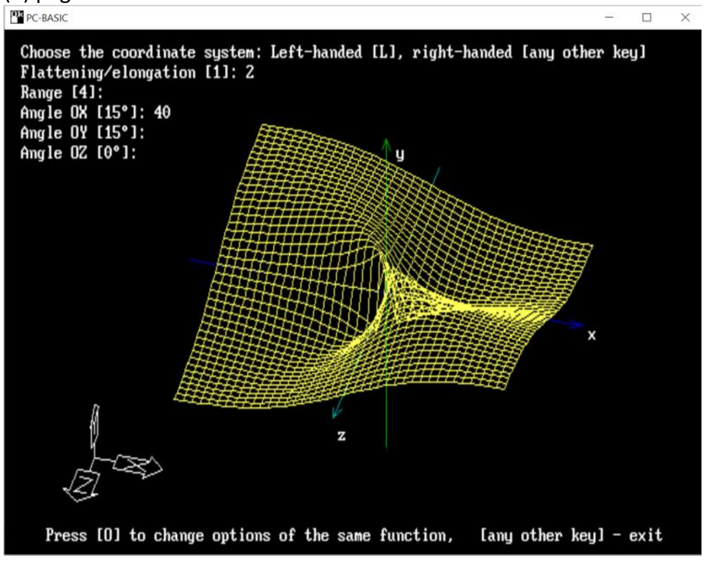
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.1
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

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

y = (xz)/(x^2+z^2)

(1) page 786



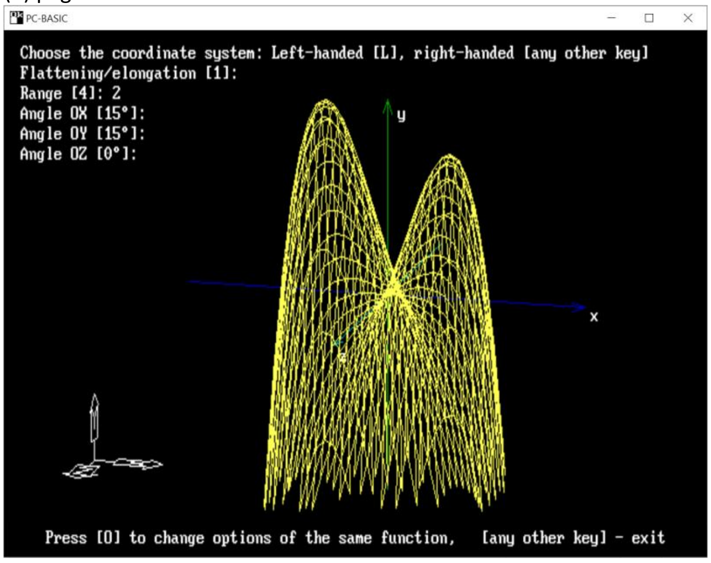
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 2
Range [4]:
Angle OX [15°]: 40
Angle OY [15°]:
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = 4*X*Z-X*X*X*X-Z*Z*Z*Z

y = 4xz-x^4-z^4

(1) page 836



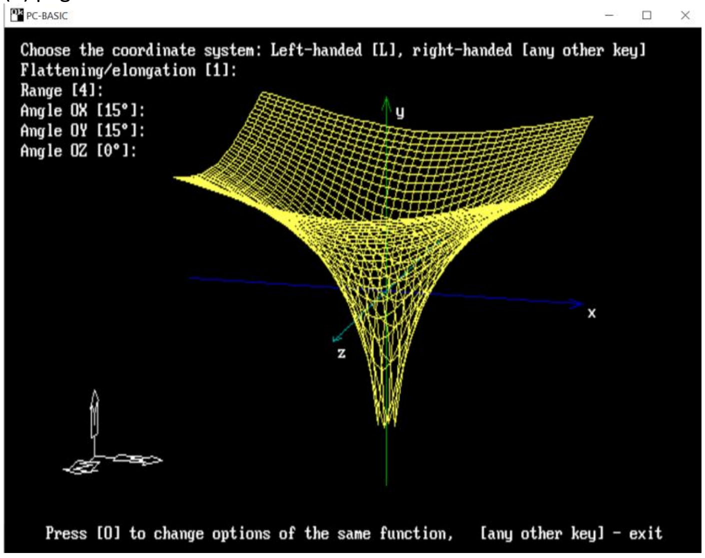
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]: 2
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = LOG(X*X+Z*Z)

y = log(x^2+z^2)

(1) page 787



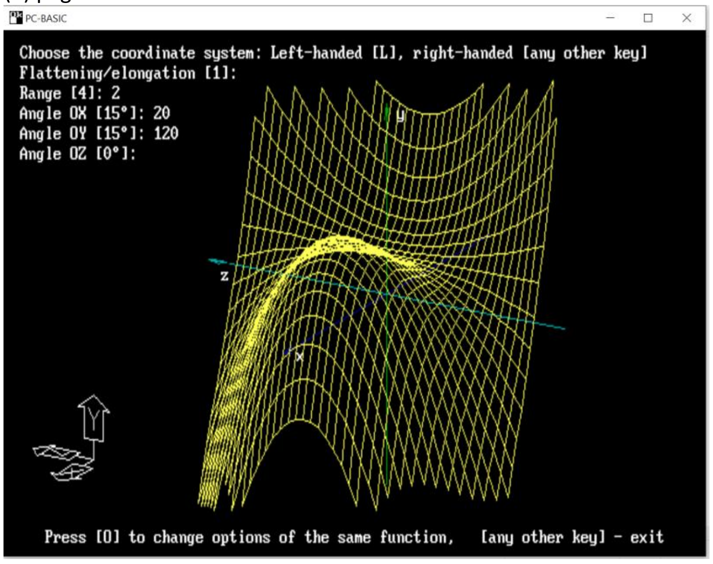
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

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



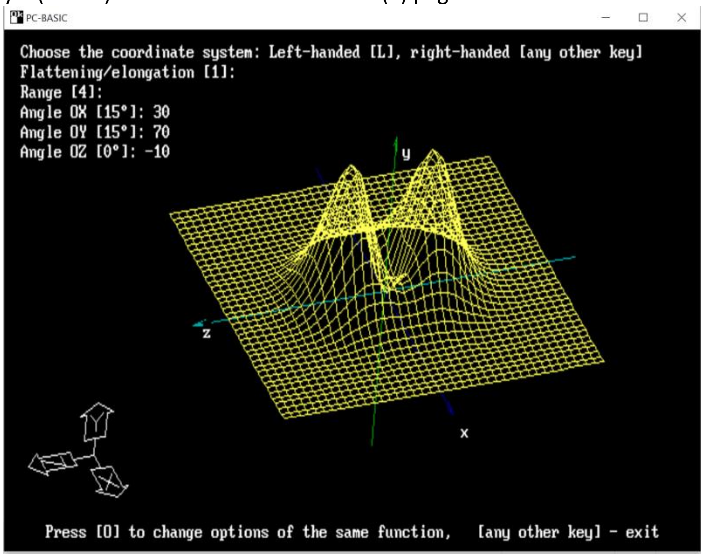
```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]: 2
Angle OX [15°]: 20
Angle OY [15°]: 120
Angle OZ [0°]:
```

```
Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = (X*X+4*Z*Z)*E^(1-X*X-Z*Z)

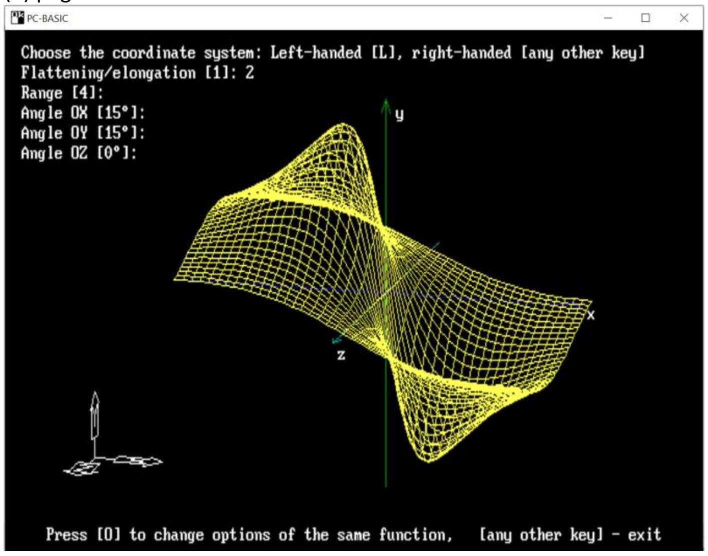
$y = (x^2+4z^2)e^{1\text{-}xx\text{-}zz}$ (1) page 836


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]: 30
Angle OY [15°]: 70
Angle OZ [0°]: -10

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = (-4*X)/(X*X+Z*Z+1)

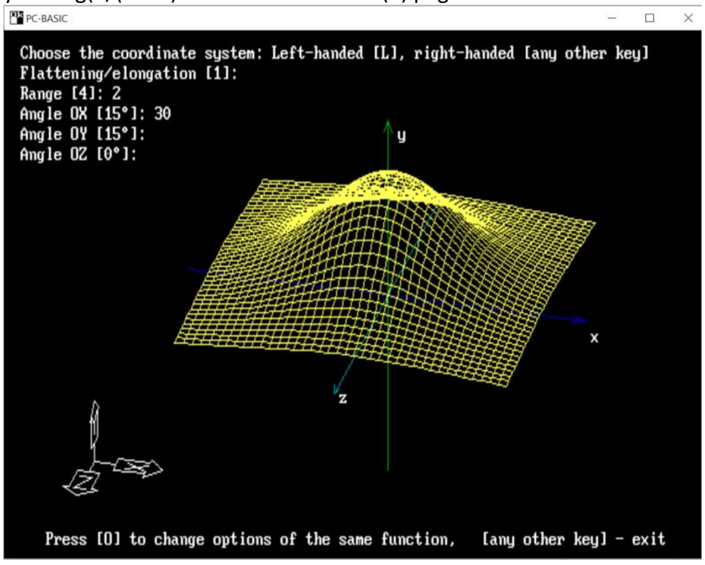
$y = -4x/(x^2+z^2+1)$

(1) page 837


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 2
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

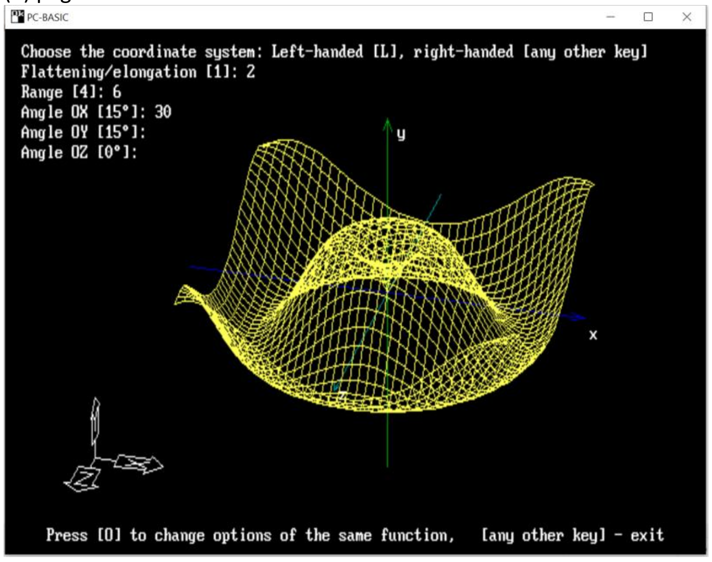
13000 Y = ATN(1/(X*X+Z*Z))

$y = \text{arctg}(1/(x^2+z^2)$ (1) page 837


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]: 2
Angle OX [15°]: 30
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = SIN(SQR(X*X+Z*Z))

(2) pages 758 and 759

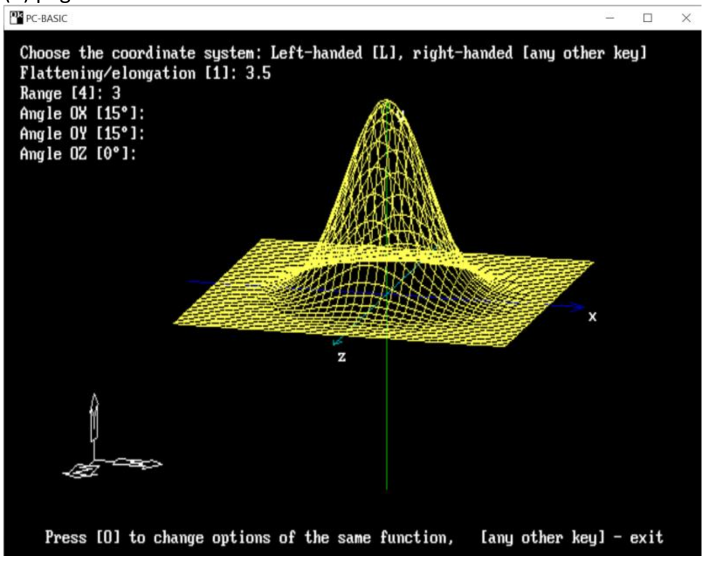

```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 2
Range [4]: 6
Angle OX [15°]: 30
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = E^(-(X*X+Z*Z))

$y = e^{-(xx+zz)}$

(1) page 837

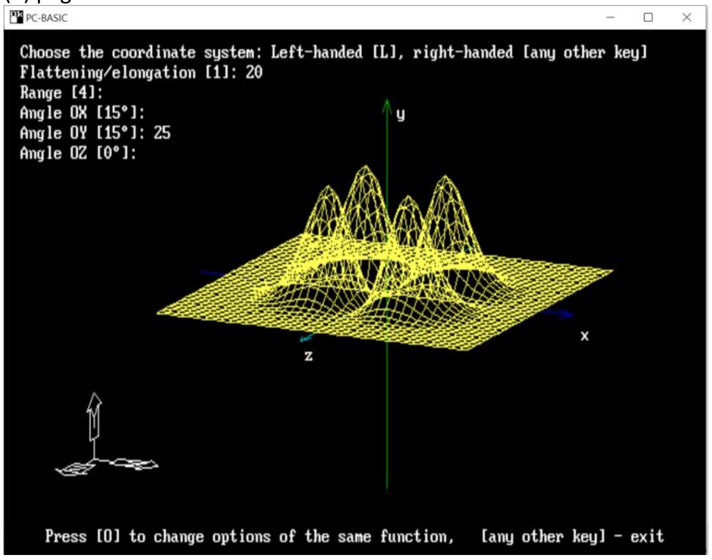

```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 3.5
Range [4]: 3
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = X*X*Z*Z*E^(-X*X-Z*Z)

$y = x^2z^2e^{-xx-zz}$

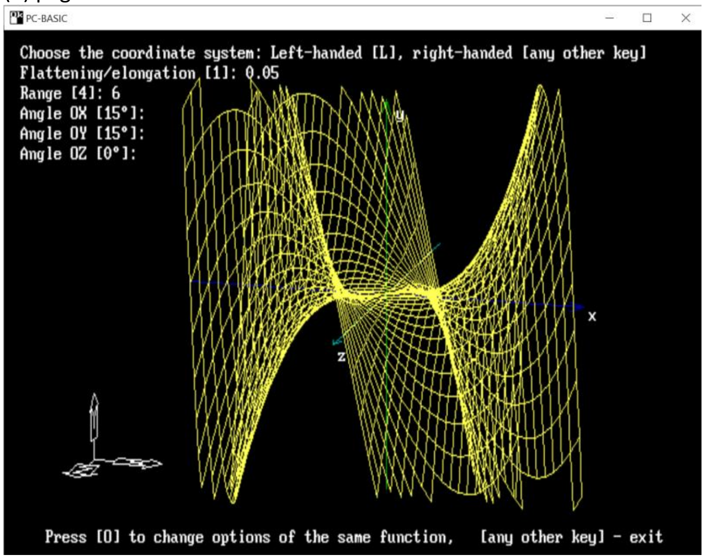
(2) pages 758 and 759


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 20
Range [4]:
Angle OX [15°]:
Angle OY [15°]: 25
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = X*X*X-3*X*Z*Z

$y = x^3 - 3xz^2$

(2) pages 758 and 759

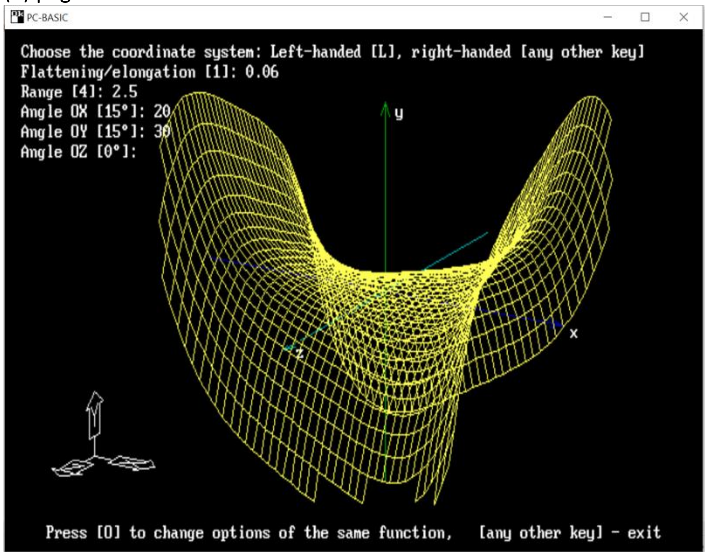


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.05
Range [4]: 6
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = X*X*X*X - Z*Z*Z*Z - 4*X*Z + 1

$y = x^4 - z^4 - 4xz + 1$

(2) page 813

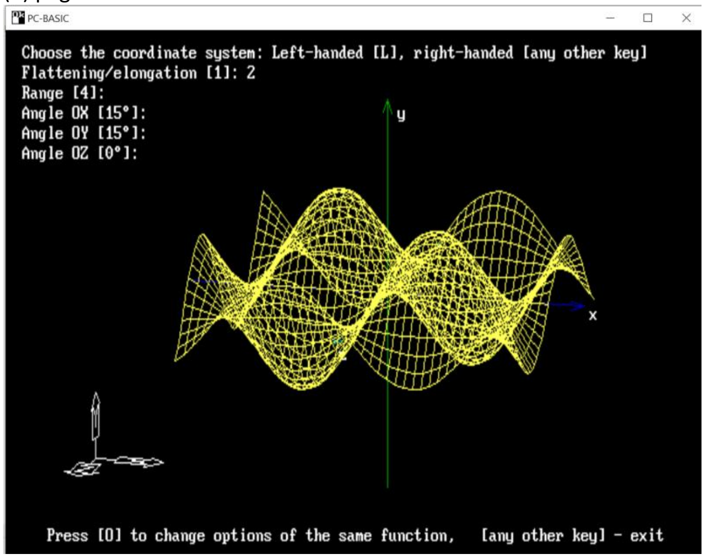


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.06
Range [4]: 2.5
Angle OX [15°]: 20
Angle OY [15°]: 30
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

y = sin x * sin z

(2) pages 758 and 759

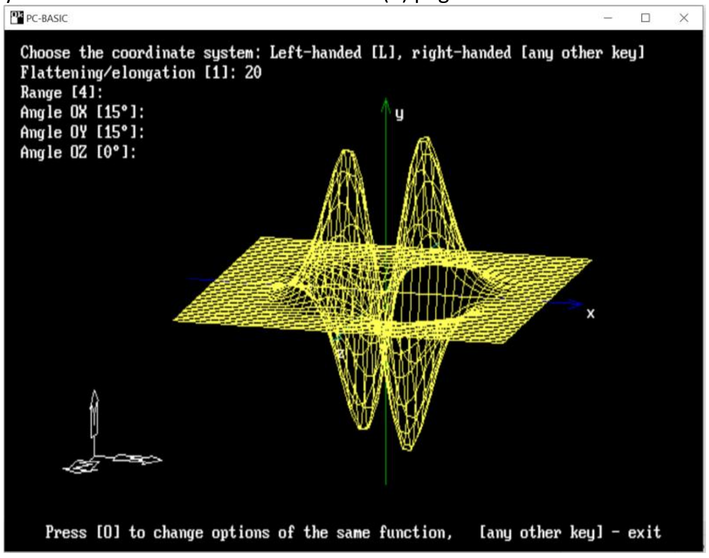


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 2
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

E = 2.71828 <-- line 194

13000 Y = -X*Z*E^(-X*X-Z*Z)

$y = -xze^{-x*x-z*z}$ (2) page 754

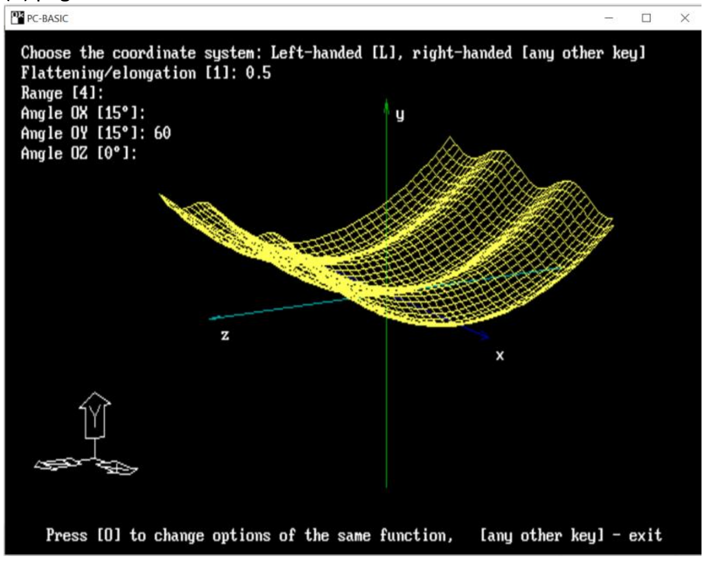


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 20
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

13000 Y = (SIN(X))^2 + 0.25*Z*Z

$y = \sin^2 x + 1/4\, z^2$

(2) pages 758 and 759

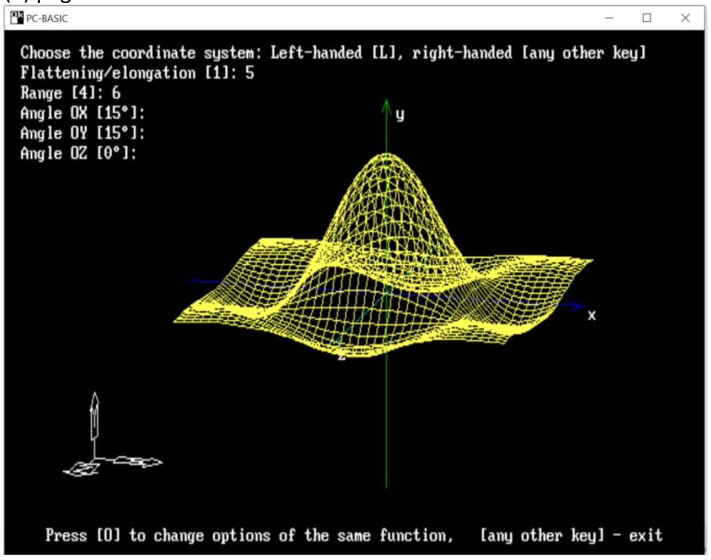


```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.5
Range [4]:
Angle OX [15°]:
Angle OY [15°]: 60
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

y = sin x sin z / (xz)

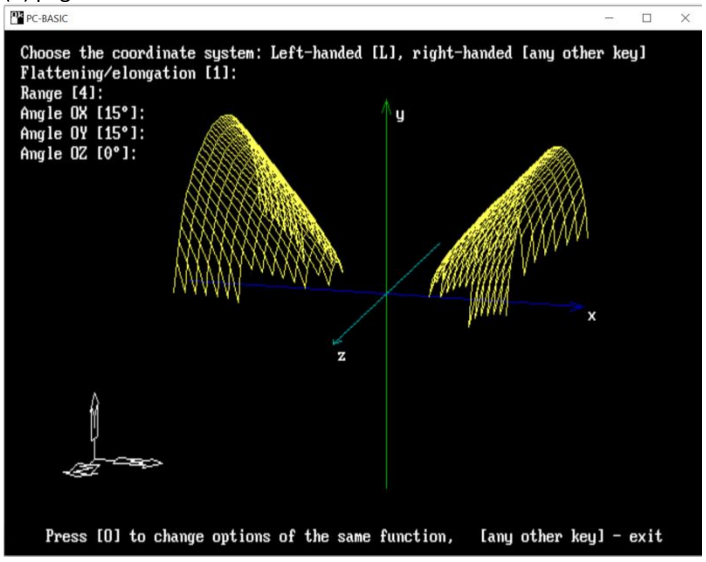
(2) page 688



```
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 5
Range [4]: 6
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [0] to change options of the same function,   [any other key] - exit
```

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

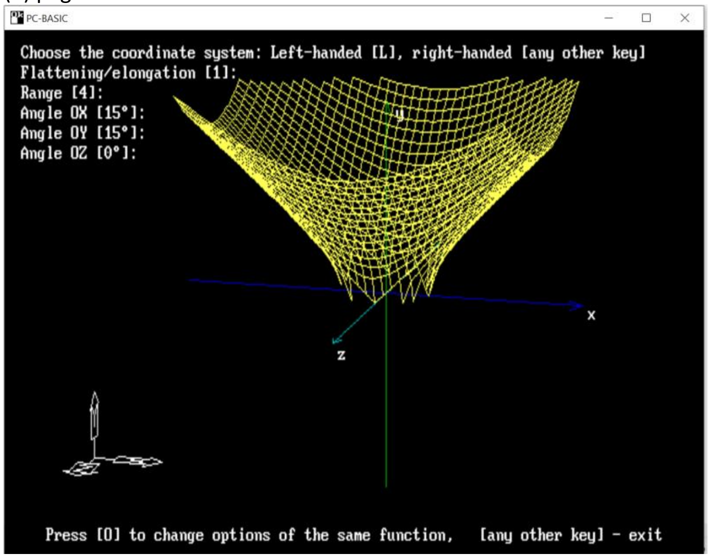
(3) page 636



```
PC-BASIC
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

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

(3) page 635

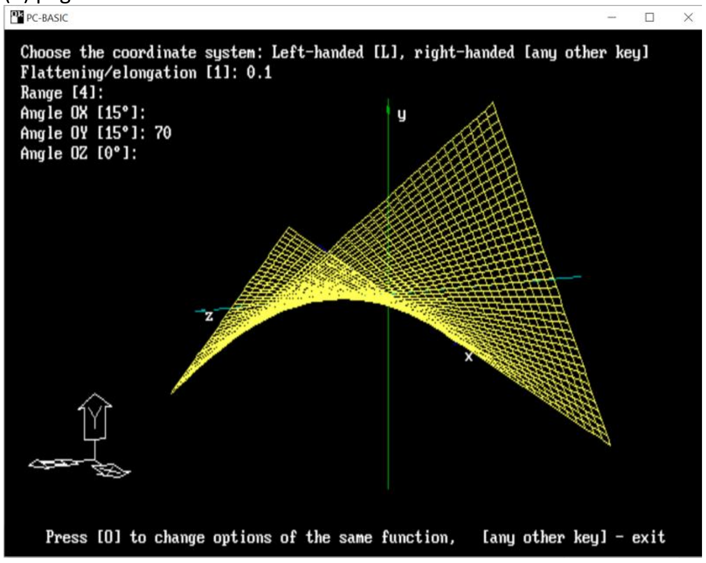


```
PC-BASIC
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]:
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

13000 Y = 2*X*Z <-- saddle

y = 2xz

(3) page S-141

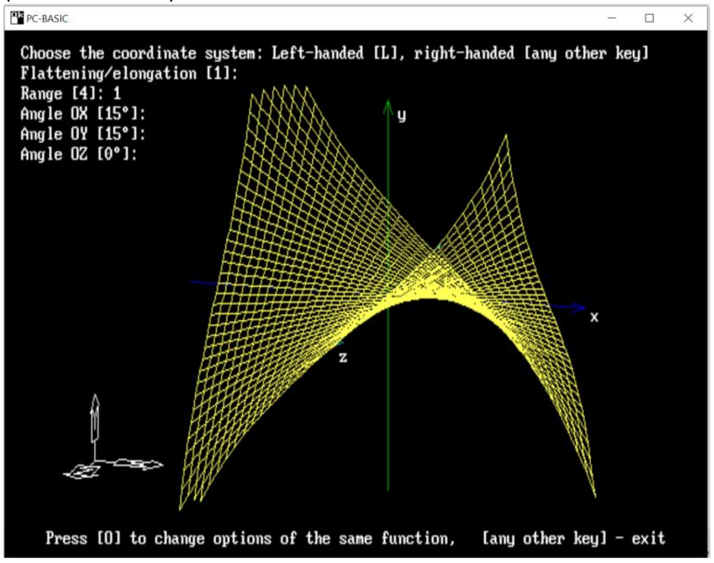


```
PC-BASIC
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]: 0.1
Range [4]:
Angle OX [15°]:
Angle OY [15°]: 70
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

13000 Y = TAN(X*Z)

y = tan(xz)

(not from a book)



```
PC-BASIC
Choose the coordinate system: Left-handed [L], right-handed [any other key]
Flattening/elongation [1]:
Range [4]: 1
Angle OX [15°]:
Angle OY [15°]:
Angle OZ [0°]:

Press [O] to change options of the same function,   [any other key] - exit
```

- - - - - - - - - - - - - - - - - - -

(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

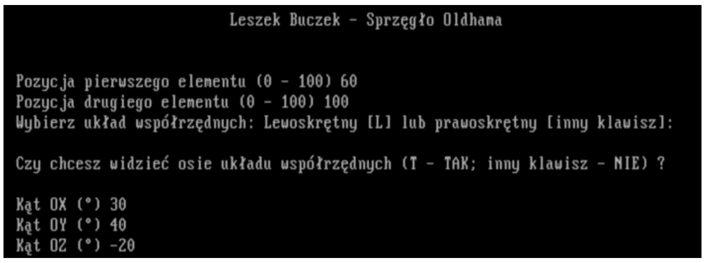
(2) James Stewart: Calculus, Concepts and Contexts; McMaster University; 1998, Brooks Cole Publishing Company, ISBN: 0-534-34330-9

(3) Sherman K. Stein: Calculus and Analytic Geometry; University of California, Davis; Fourth edition; 1987, McGraw-Hill Book Company, ISBN: 0-07-061159-9

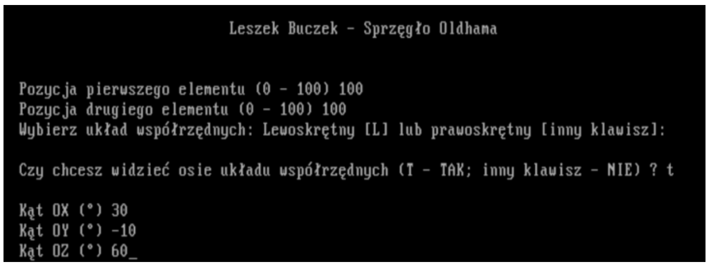
(4) Harley Flanders: Calculus; Florida Atlantic University; 1985, W. H. Freeman and Company, ISBN: 0-7167-1643-7

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

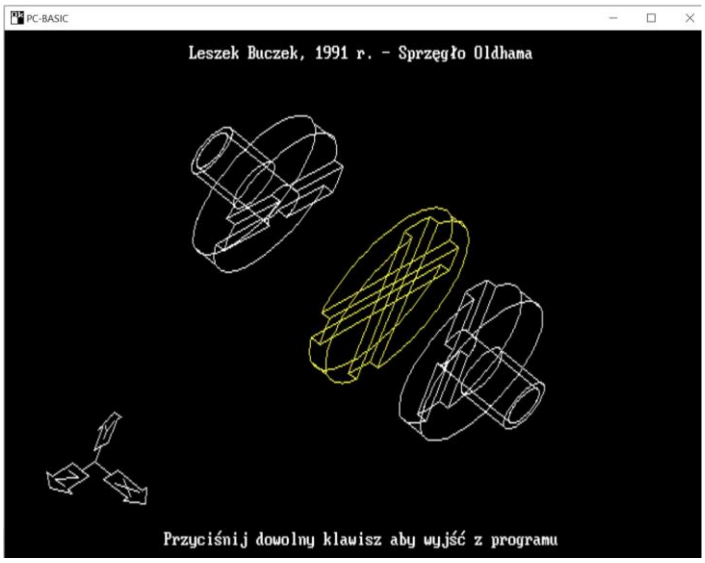
Polish	English
Sprzegło Oldhama	Oldham coupling
Pozycja pierwszego elementu (0 - 100)	Position of the first element (0 - 100)
Pozycja drugiego elementu (0 - 100)	Position of the second element (0 - 100)
Wybierz układ współrzędnych: Lewoskrętny [L] lub prawoskrętny [inny klawisz]:	Select the coordinate system: Left-hand [L] or right-hand [other key]:
Czy chcesz widzieć osie układu współrzędnych (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 (°)
Przyciśnij dowolny klawisz aby wyjść z programu	Press any key to exit the program



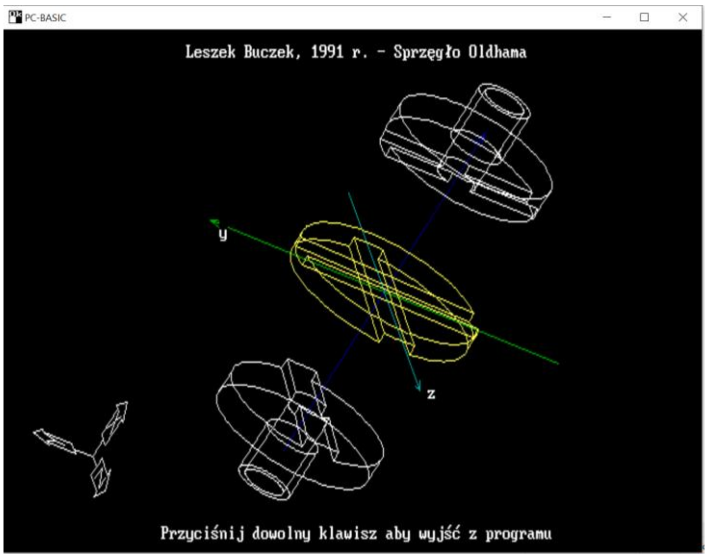
```
                    Leszek Buczek - Sprzęgło Oldhama


Pozycja pierwszego elementu (0 - 100) 60
Pozycja drugiego elementu (0 - 100) 100
Wybierz układ współrzędnych: Lewoskrętny [L] lub prawoskrętny [inny klawisz]:

Czy chcesz widzieć osie układu współrzędnych (T - TAK; inny klawisz - NIE) ?

Kąt OX (°) 30
Kąt OY (°) 40
Kąt OZ (°) -20
```



```
                    Leszek Buczek - Sprzęgło Oldhama


Pozycja pierwszego elementu (0 - 100) 100
Pozycja drugiego elementu (0 - 100) 100
Wybierz układ współrzędnych: Lewoskrętny [L] lub prawoskrętny [inny klawisz]:

Czy chcesz widzieć osie układu współrzędnych (T - TAK; inny klawisz - NIE) ? t

Kąt OX (°) 30
Kąt OY (°) -10
Kąt OZ (°) 60_
```



```
Leszek Buczek, 1991 r. - Sprzęgło Oldhama

Przyciśnij dowolny klawisz aby wyjść z programu
```



```
Leszek Buczek, 1991 r. - Sprzęgło Oldhama

Przyciśnij dowolny klawisz aby wyjść z programu
```

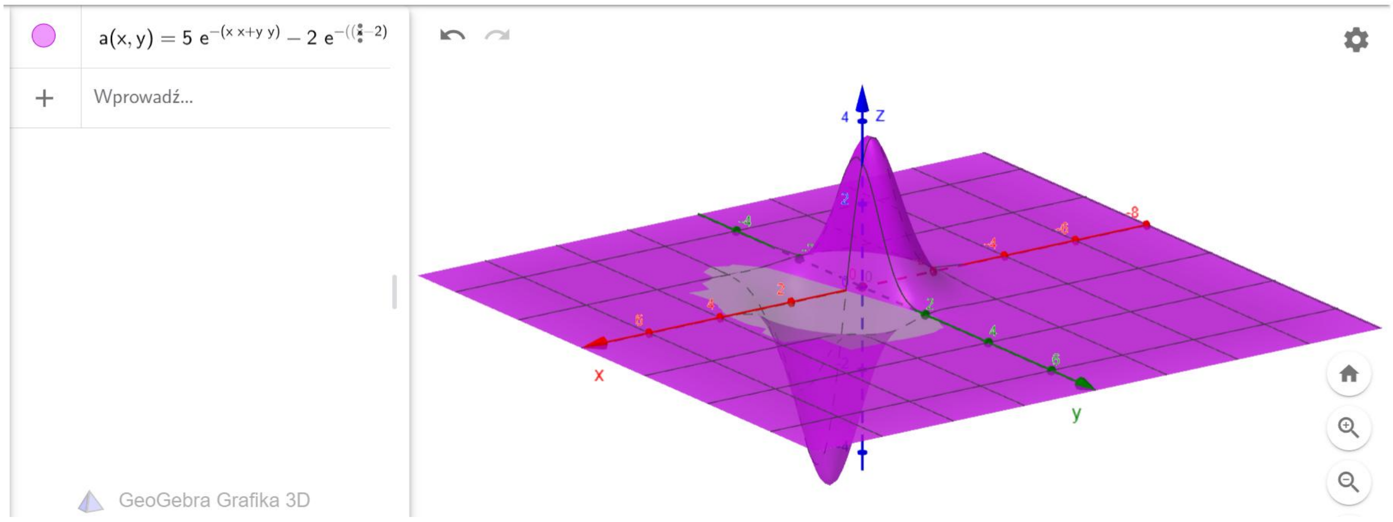
How other applications do it ?

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


$a(x, y) = 5\ e^{-(x\ x+y\ y)} - 2\ e^{-((x-2)}$

A wonderful tool for presenting 3-D graphics.

- ✓ To obtain graphics, you don't need to 'import,' 'include,' 'download' anything – you can immediately enter dependencies, not just functional ones.
- ✓ The application has many tools that greatly expand the possibilities of presenting graphical images.

Despite the fascination I experience playing with GeoGebra, it remains outside the scope of this presentation, which is intended to provide the full code in a chosen programming language for creating graphics.

What stage of development was GeoGebra at in 1991, the year I first shared my program with others? I couldn't do it for everyone that time, so I am doing it now.

How *Python* does it ?

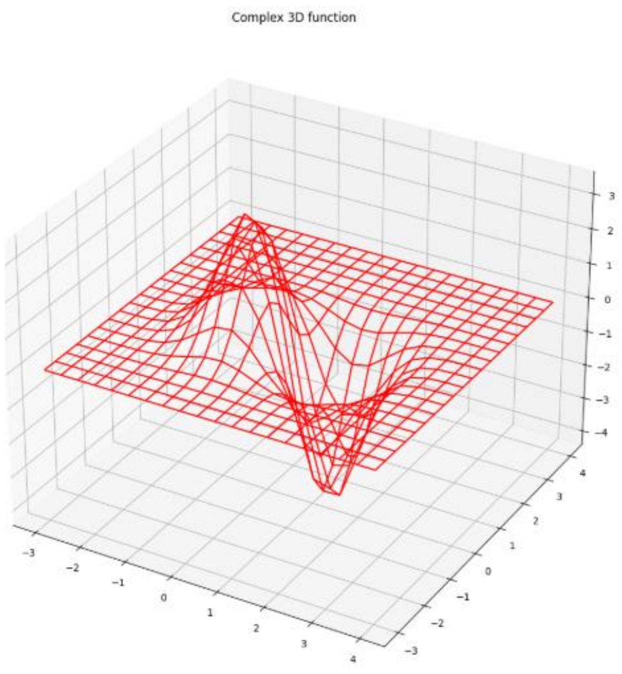
To obtain an image of a spatial figure in any of its settings, you need to import such miraculous mechanisms as:
- OpenGL.GL/GLU/GLUT
- vpython
or something similar.

If I give them up, what is left to me is the *matplotlib* with *numpy*. The result is good. However, 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+Y*Y)
# VV=-((X-2)*X+(Y-.5)*Y)
# Z=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('Complex 3D function');
```



Such things does even Excel.

What is the easiest way to make such a one-sided presentation of a spatial figure?

Let's assume we want to execute the function `y=f(x,z)` with the **X** and **Y** axes on the screen as if there were no **Z** axis, but it is: perpendicular to the screen.
Any numbers concerning the displacement of a specific point by the number of pixels on the screen are only examples.

Let's start by drawing a broken curve for z=-10.
And calculate **y** as **f(x,z)**, where **z**=-10 and **x** varies from -10 to 10 in increments of 1. We will obtain 21 points on the screen. During these 21 calculations:

- we store the pixel position of each point in the `DIM POSX(21)` and `DIM POSY(21)` arrays (the names suggest: position of x and position of y),
- we connect these points one by one using the `LINE -( POSX(i), POSY(i) )` instruction, obtaining a broken line made of the connected 21 points.

Now we perform identical calculations for **z**=-9, however:

- We **shift each calculated pixel position: 5 pixels down and 3 pixels to the right.**
- Immediately after calculating position of each pixel:
  - We connect it to the corresponding point `(POZX(i),POZY(i))` previously calculated for **z**=-10,
  - We change `POZX(i)` and `POZY(i)` from those calculated for **z**=-10 to the values for **z**=-9 (to prepare data for the next line, i.e. for **z**=-8),
  - We connect each point calculated for **z**=-9, just as we did earlier for **z**=-10. We get a sequence of 21 small trapezoids (unless the lines intersect).

(At this point, you can try filling each trapezoid with a color, every other one with a different color—preferably a different shade, e.g., light gray and dark gray. Repeating the colors of the next line of trapezoids in this way — for **z** increased by 1 — will result in covering the invisible lines and trapezoids with new lines and trapezoids (hence from -10 to 10 and not the other way around). This task may not be so simple, as the new points may be either below or above the old points on the screen. Furthermore, the new lines may intersect with the old ones.)

Once we reach the stage where we finish drawing the broken curve for **z** = 10, we consider the graph complete.

If you want this graph to be skewed, like the last Python result, then for the next '**y**' result, raise the point by another two pixels. Repeat this for each '**z**' separately, raising the calculated '**y**' by another two pixels for the next '**x**'.

As a result, we get is good, but it is only an overall view of the spatial function and it has little to do with rotations of the figure by arbitrary angles. It does not require any knowledge of mathematics. This is a task for a 14-year-old IT enthusiast.