

## Funkcja 3-D

To mój pierwszy program na PC z 1991 roku. Wykonanie poniższej grafiki zajęło procesorowi Intel 80283 pełne dwie minuty. Program wysłałem w tym samym roku do czasopisma *Bajtek* (dodałem przykład rysunku sprzęgła Cardana). Odebrano go, ale nie publikowano. Była to moja reakcja na nieudolne próby tworzenia grafiki 3-D w tym czasopiśmie.

Poniżej jest obraz graficzny funkcji:

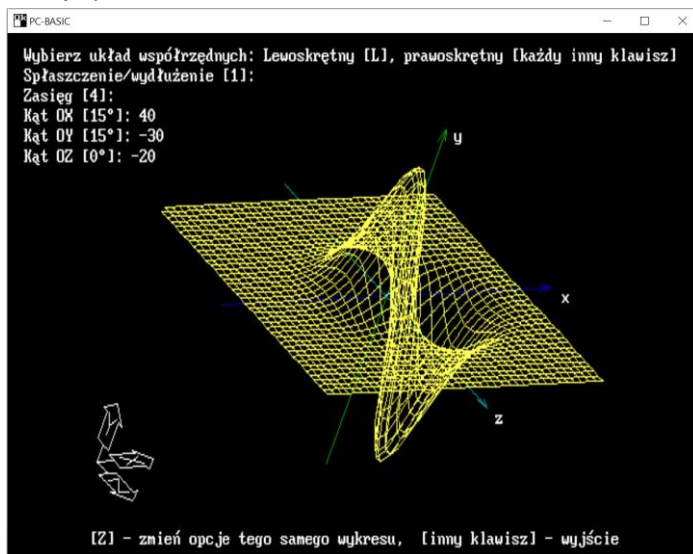
$$V = -(X * X + Z * Z)$$

$$VV = -((X - 2) * X + (Z - .5) * Z)$$

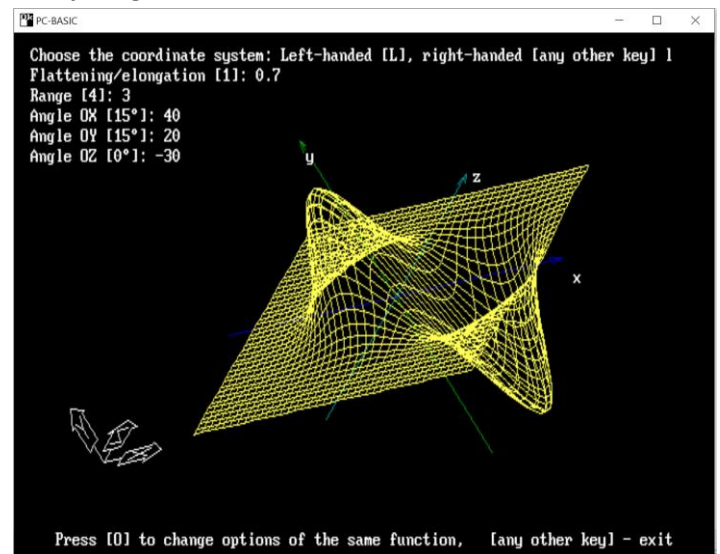
$$Y = 5 * \text{EXP}(V) - 2 * \text{EXP}(VV)$$

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

Wersja polska:



Wersja angielska:



Niestety proces obsługi błędów nie jest w GWBasic'u dopracowany. Podaję dwie wersje kodu spodziewając się takich samych wyników:

```
10  <-- linia FOR poniżej - dwie wersje
20  PRINT "I=" I, "log(i)=" LOG(I);
30  ON ERROR GOTO 60
40  PRINT "      Nie ma błedu (No error)"
50  GOTO 90
60  A=ERR: B=ERL
70  PRINT: PRINT "Numer błedu (Error number)=" A "      Linia błedu (Error line)=" B
80  RESUME 90
90  NEXT
100 PRINT: PRINT "Proces zakończony pomyślnie (Process completed successfully)"
110 END
```

Pierwsza wersja:

```
10 FOR I=2 TO -2 STEP -1          <---->
Wynik:
I= 2      log(i)= .6931472      Nie ma błedu (No error)
I= 1      log(i)= 0             Nie ma błedu (No error)
I= 0      log(i)=
Numer błedu (Error number)= 5    Linia błedu (Error line)= 20
I=-1      log(i)=
Numer błedu (Error number)= 5    Linia błedu (Error line)= 20
I=-0      log(i)=
Numer błedu (Error number)= 5    Linia błedu (Error line)= 20
```

Druga wersja:

```
10 FOR I=-2 TO 2 STEP 1
Wynik:
I=-2      log(i)=
Illegal function call in 20
OK
```

Process zakończony pomyślnie (Process completed successfully)  
OK

Bardzo dobrze - wszystko jak oczekiwano

Bardzo źle - nieoczekiwany rezultat !

Przy wpisaniu wyrażenia algebraicznego funkcji nie można liczyć na przypadek poprawnego procesu przetwarzania błędów. Dlatego też koniecznym jest wpisanie tuż nad linią funkcji  $f(x,z)$  linii omijających niewykonalne operacje matematyczne (pokolorowane linie poniżej), poprzez:

**IF ( warunek do ominięcia ) THEN BLAD=1: GOTO 14050**

Przykład:

```
12090 IF (X+Z <= 0) THEN BLAD=1: GOTO 14050 ' ta linia omija błąd
13000 Y=LOG(X+Z) ' <-- tu już błędu nie będzie
```

Inny przykład:

```
12080 IF (X-2*Z <= 0) THEN BLAD=1: GOTO 14050 ' ta linia omija błąd
12090 IF (X + LOG(X-2*Z) < 0) THEN BLAD=1: GOTO 14050 'linia omija błąd
13000 Y= 2 + SQRT(X + LOG(X-2*Z)) ' <-- tu już błędu nie będzie
```

Koniecznym jest tu podanie BLAD=1 informujący dalszą część procesu, że wystąpiła niepoprawna operacja matematyczna.

Jestem przekonany, że jest jakaś metoda wychwycenia błędu nr 5 czyli 'illegal function call' ale musi ona jakoś sztucznie oszukiwać program. Z błędem nr 11 czyli 'Division by zero' (dzielenie przez zero) nie ma tego kłopotu.

Należy pamiętać, że każda wartość spoza zakresu liczb całkowitych (od -32768 do 32767) powoduje błąd "OVERFLOW" i wstrzymanie działania programu.

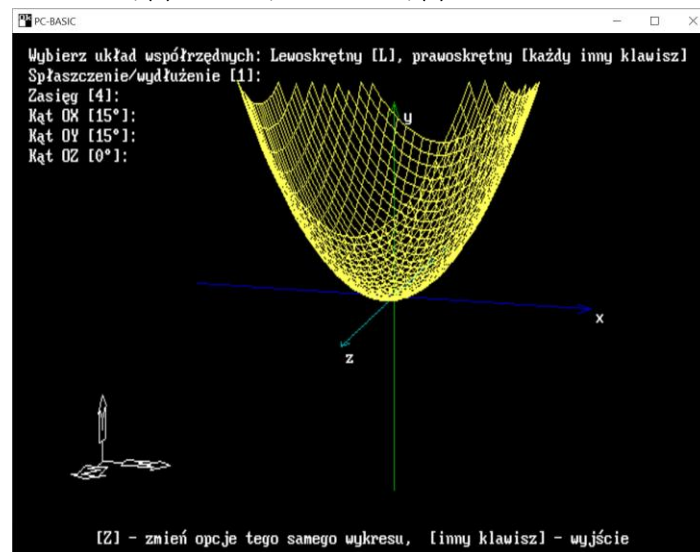
-----

W poniższych przykładach piszę  $X*X*X$  chociaż zawsze można napisać  $X^3$

**13000 Y = X \* X / 2 + Z \* Z / 6**

$y = x^2/a^2 + z^2/b^2$  Elliptic paraboloid

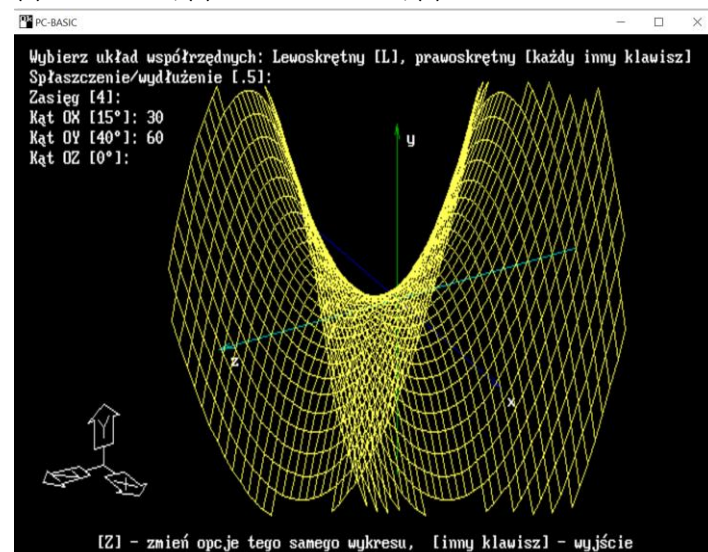
str. 716 i 719, (2)  $Y=4x^2+z^2$ , str. 687 i 688, (4) str. 756



**13000 Y = Z \* Z - X \* X**

$y = z^2/b^2 - x^2/a^2$  Hyperbolic paraboloid

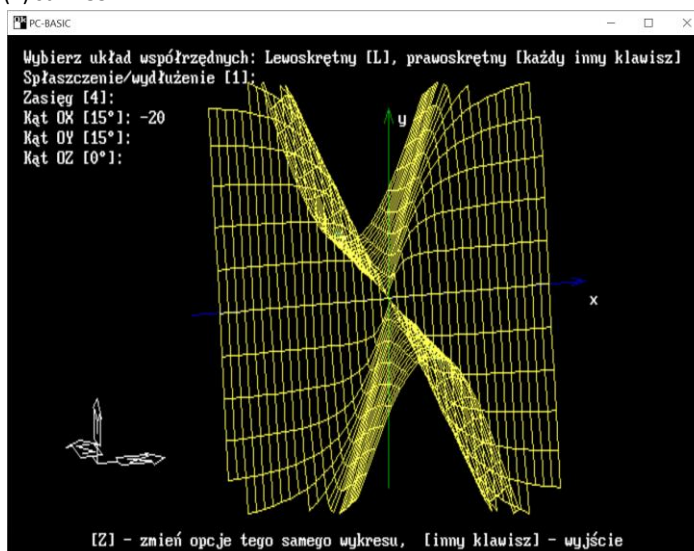
(1) str. 716 i 775, (2) str. 687 i 688 + 812, (4) str. 756 i 801



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

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

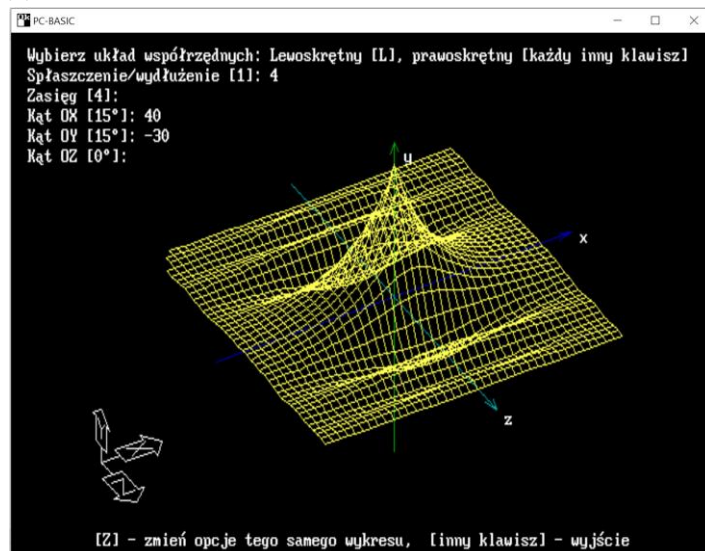
(1) str. 785



$$E = 2.71828 \quad <-- \text{ linia 194}$$

$$13000 Y = \cos(Z * Z) * E^{(-\sqrt{X * X + Z * Z})}$$

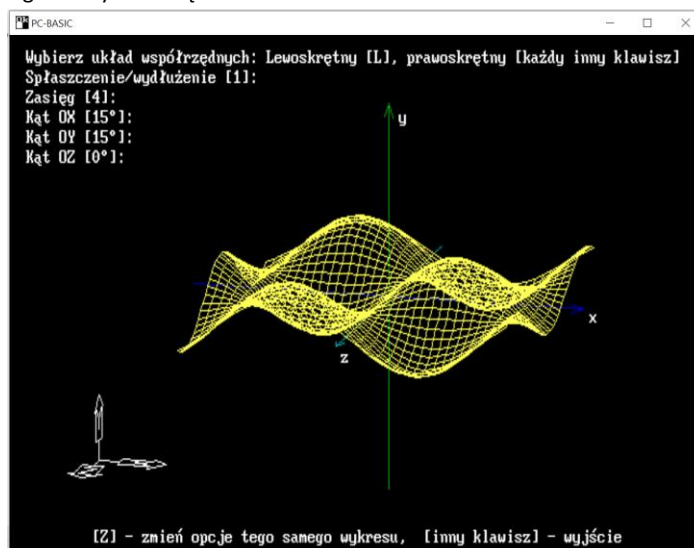
(1) str.783



$$13000 Y = \sin(X) * \cos(Z)$$

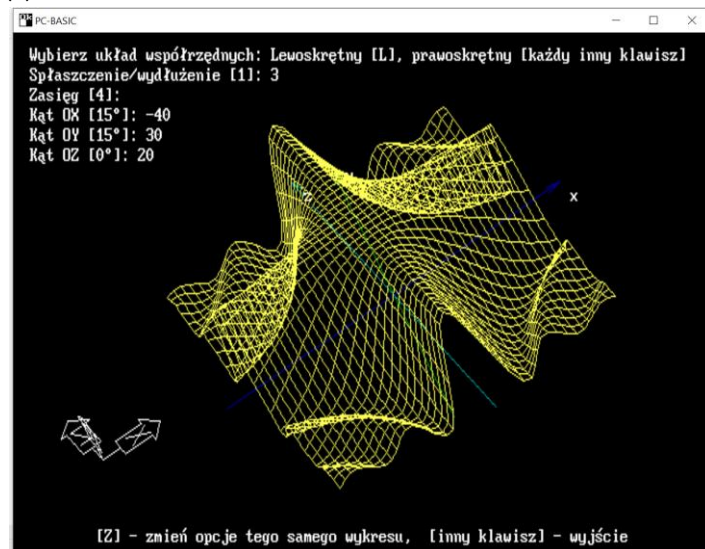
$$y = \sin x * \cos z$$

tego nie było w książkach



$$13000 Y = \sin(X * Z) / (X * Z)$$

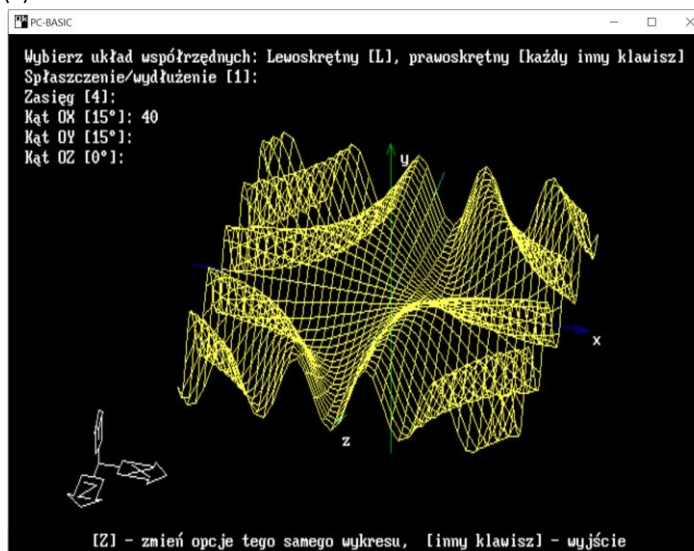
(1) str.785



$$13000 Y = \sin(X * Z)$$

$$y = \sin xz$$

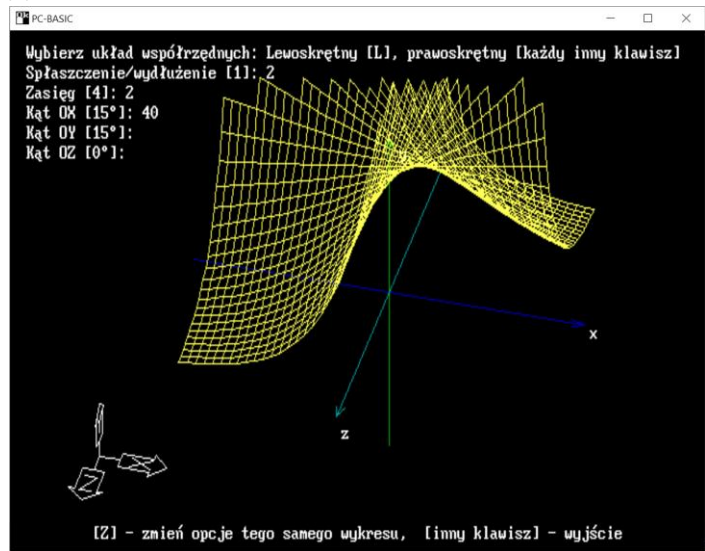
(1) str 783



$$E = 2.71828 \quad <-- \text{ linia 194}$$

$$13000 Y = E^{(X * Z)}$$

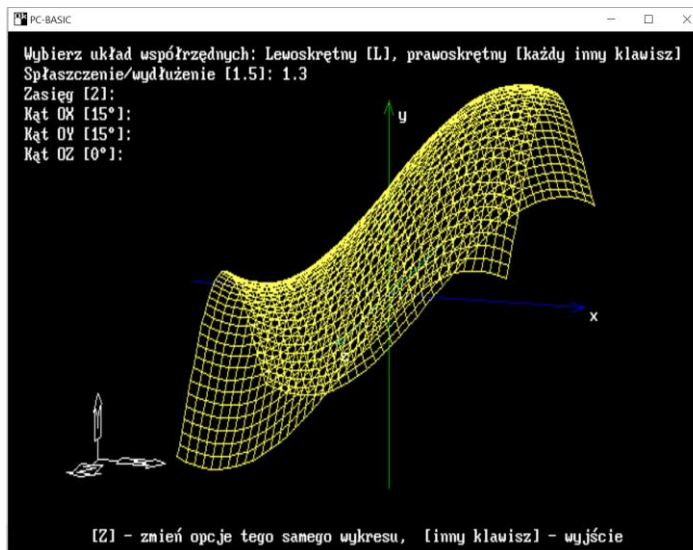
(1) str.786





$$13000 \ Y = \sin(X) + \cos(Z)$$

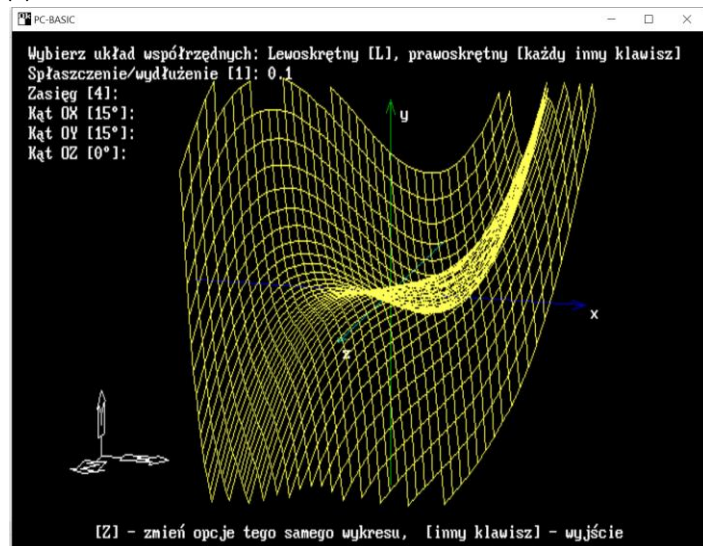
(1) str.786



$$13000 \ Y = X^3 - 3XZ + Z^3$$

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

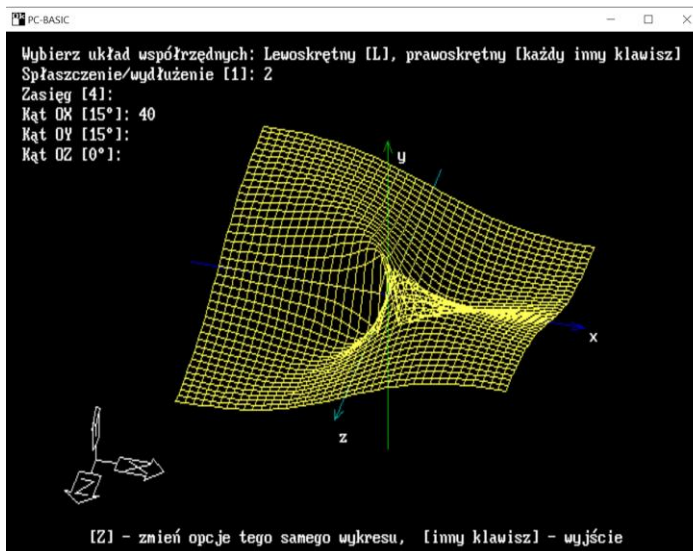
(1) str.836



$$13000 \ Y = (X \cdot Z) / (X^2 + Z^2)$$

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

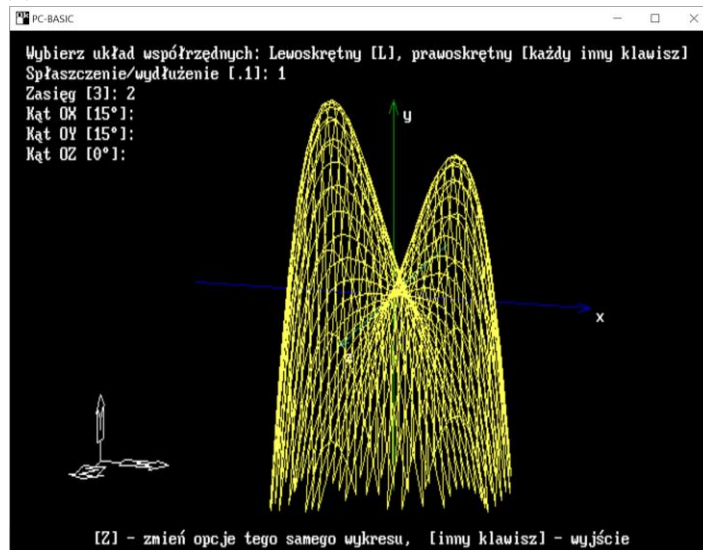
(1) str.786



$$13000 \ Y = 4 \cdot X \cdot Z - X^4 - Z^4$$

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

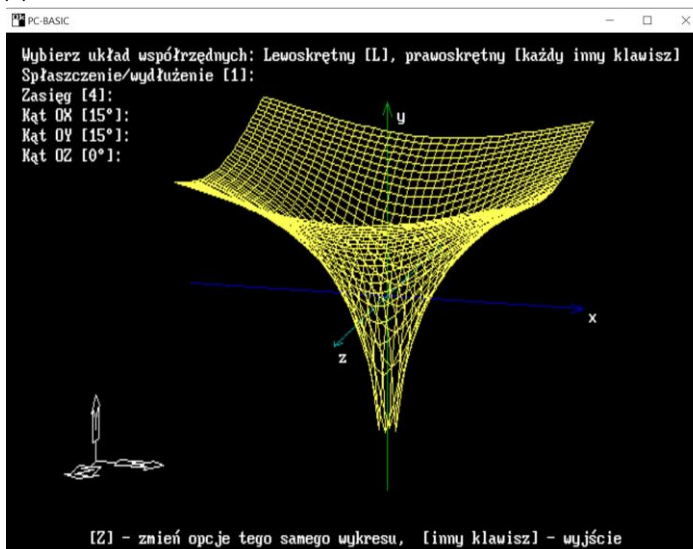
(1) str.836



$$13000 \ Y = \log(X^2 + Z^2)$$

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

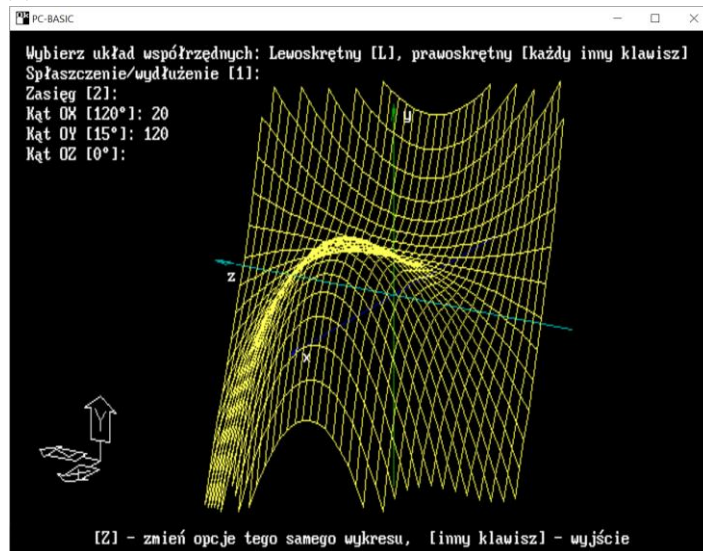
(1) str.787



$$13000 \ Y = (3 \cdot X^2 + 1) / 2 - X \cdot (X^2 + Z^2)$$

$$y = (3x^2 + 1) / 2 - x(x^2 + z^2)$$

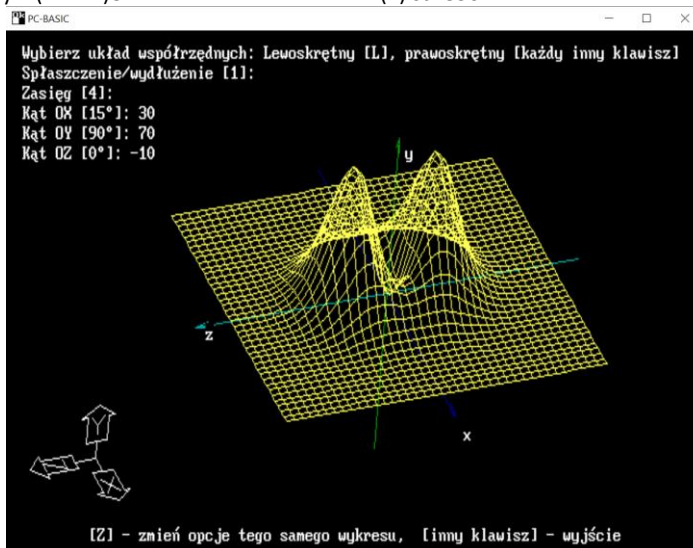
(1) str.836



E = 2.71828 <-- linia 194

$$13000 Y = (X^2 + 4Z^2) * E^{(1 - X^2 - Z^2)}$$

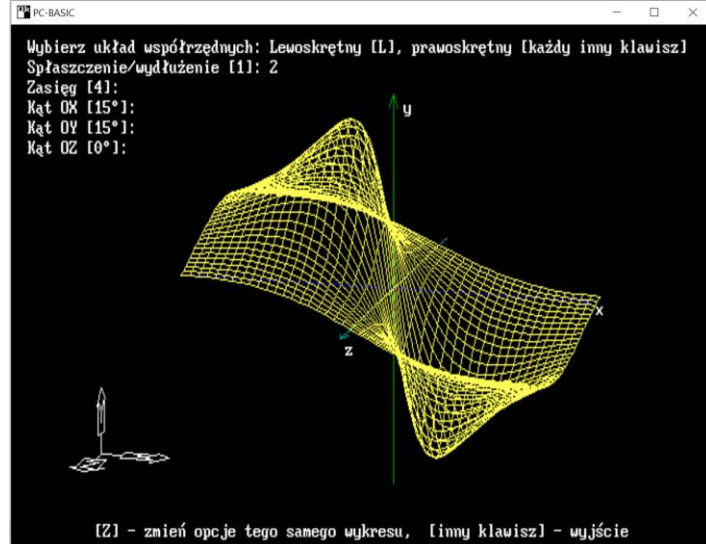
$$y = (x^2 + 4z^2) e^{1 - x^2 - z^2} \quad (1) \text{ str.836}$$



$$13000 Y = (-4 * X) / (X^2 + Z^2 + 1)$$

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

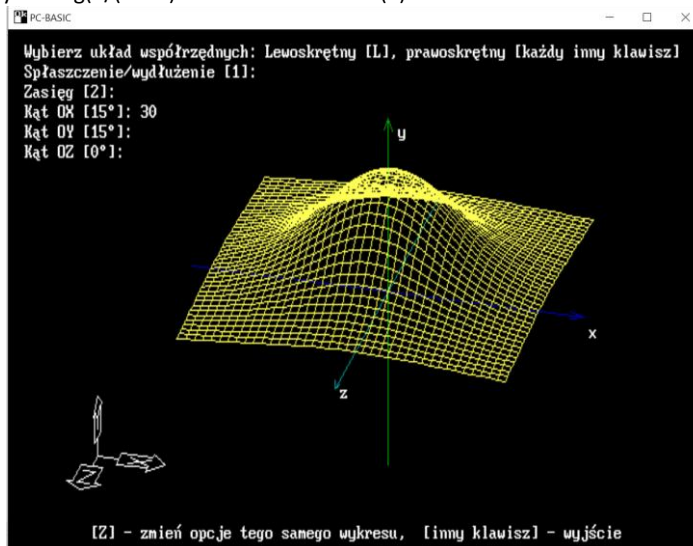
(1) str.837



$$13000 Y = \text{ATN}(1 / (X^2 + Z^2))$$

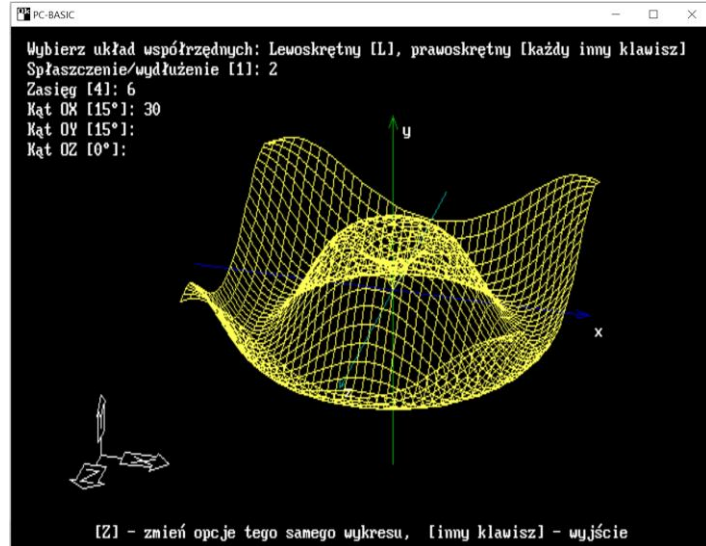
$$y = \arctg(1 / (x^2 + z^2))$$

(1) str.837



$$13000 Y = \text{SIN}(\text{SQR}(X^2 + Z^2))$$

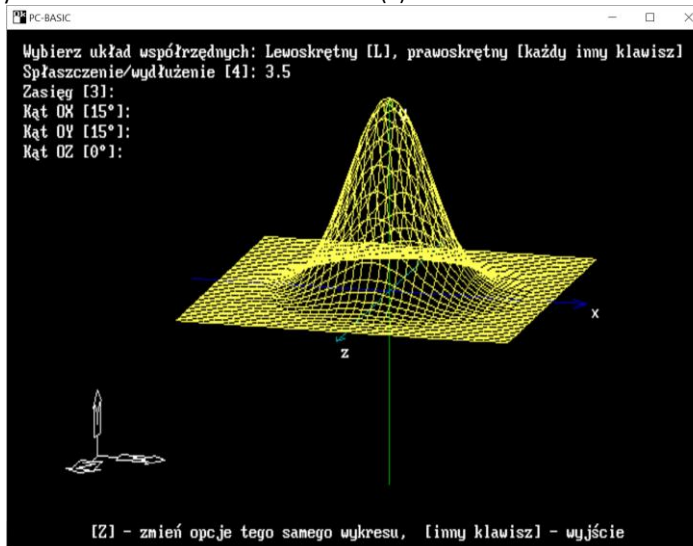
(2) str.758 i 759



E = 2.71828 <-- linia 194

$$13000 Y = E^{-(X^2 + Z^2)}$$

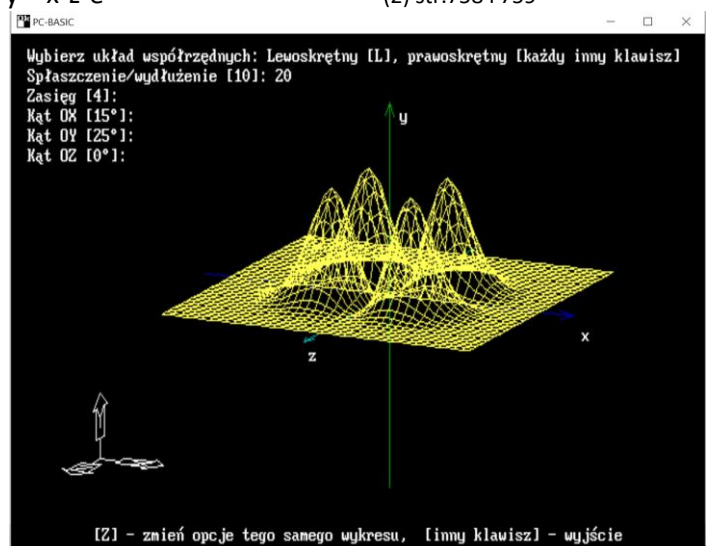
$$y = e^{-(x^2 + z^2)} \quad (1) \text{ str.837}$$



E = 2.71828 <-- linia 194

$$13000 Y = X^2 * Z^2 * E^{-(X^2 + Z^2)}$$

$$y = x^2 z^2 e^{-(x^2 + z^2)} \quad (2) \text{ str.758 i 759}$$

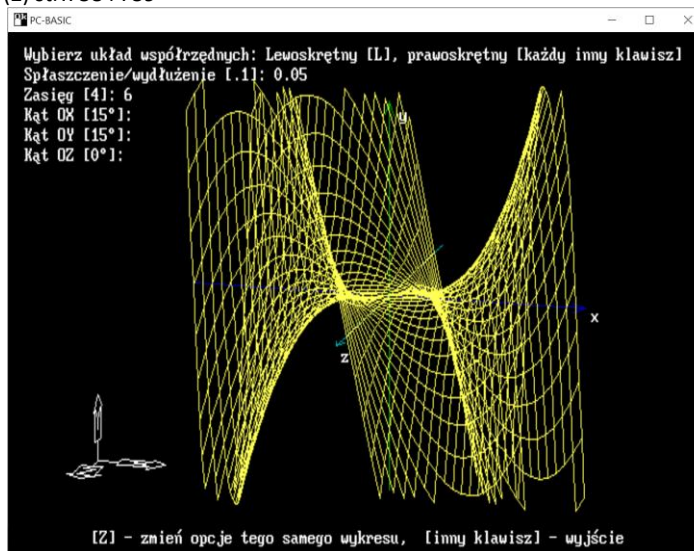




$$13000 \ Y = X^3 - 3XZ^2$$

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

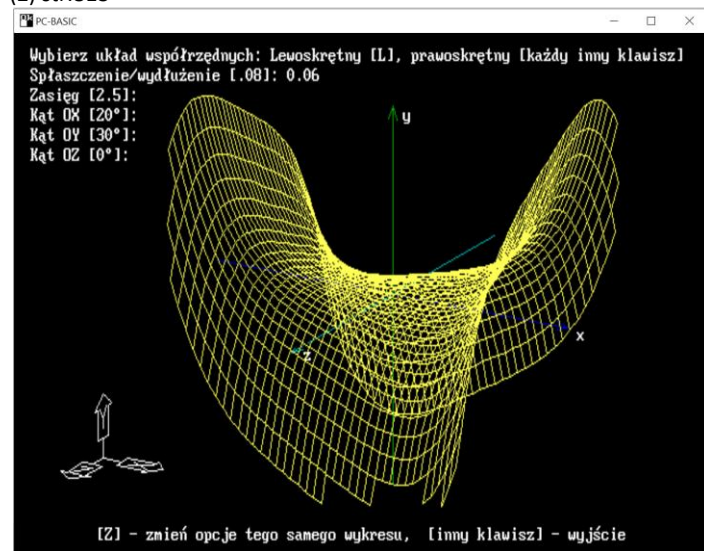
(2) str.758 i 759



$$13000 \ Y = X^4 - Z^4 - 4XZ + 1$$

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

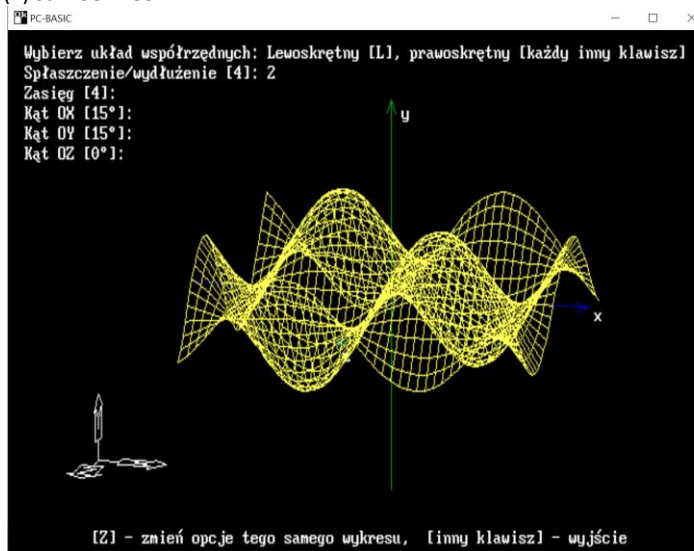
(2) str.813



$$13000 \ Y = \sin(X) * \sin(Z)$$

$$y = \sin x * \sin z$$

(2) str.758 i 759

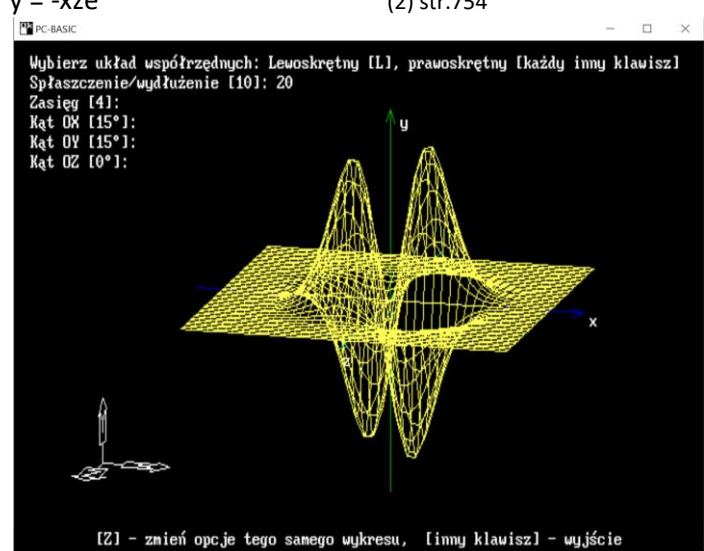


$$E = 2.71828 \quad \leftarrow \text{linia 194}$$

$$13000 \ Y = -X^2 * E^{(-X^2 - Z^2)}$$

$$y = -xz e^{-x^2 - z^2}$$

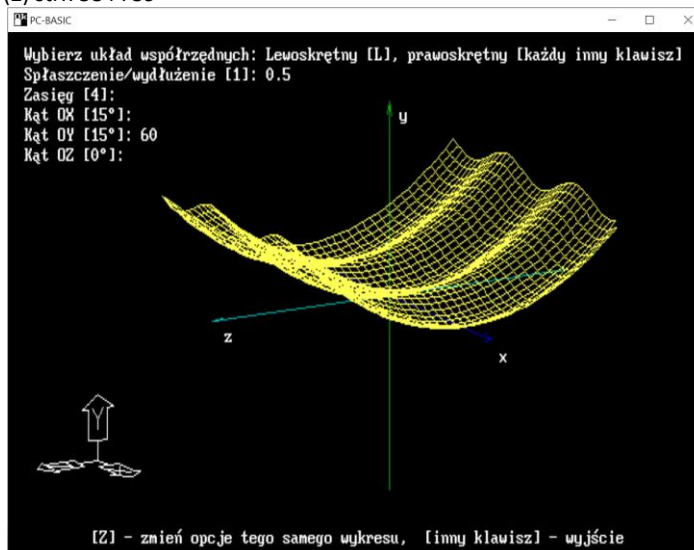
(2) str.754



$$13000 \ Y = (\sin(X))^2 + 0.25 * Z^2$$

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

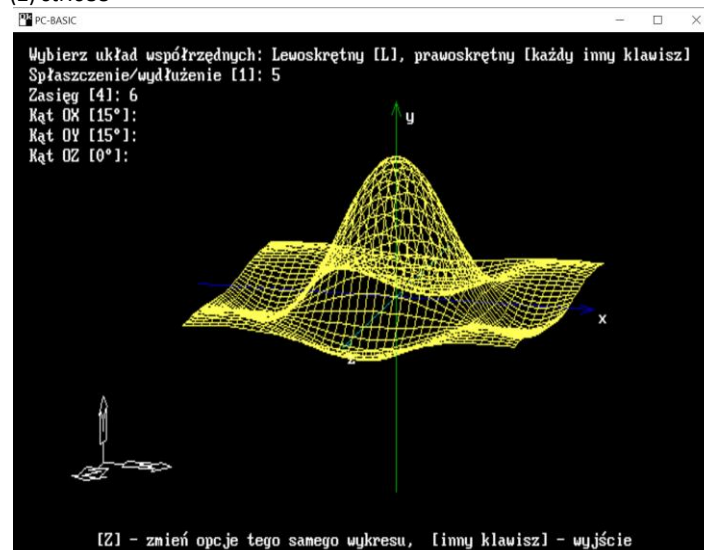
(2) str.758 i 759



$$13000 \ Y = (\sin(X) * \sin(Z)) / (X * Z)$$

$$y = \sin x \sin z / (xz)$$

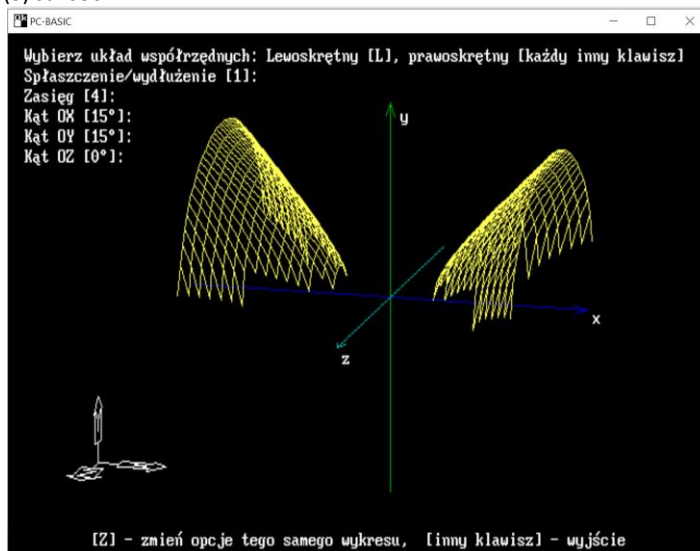
(2) str.688



12990 IF X\*X-Z\*Z-1 < 0 THEN BLAD=1: GOTO 14050

13000 Y = SQR(X\*X-Z\*Z-1)

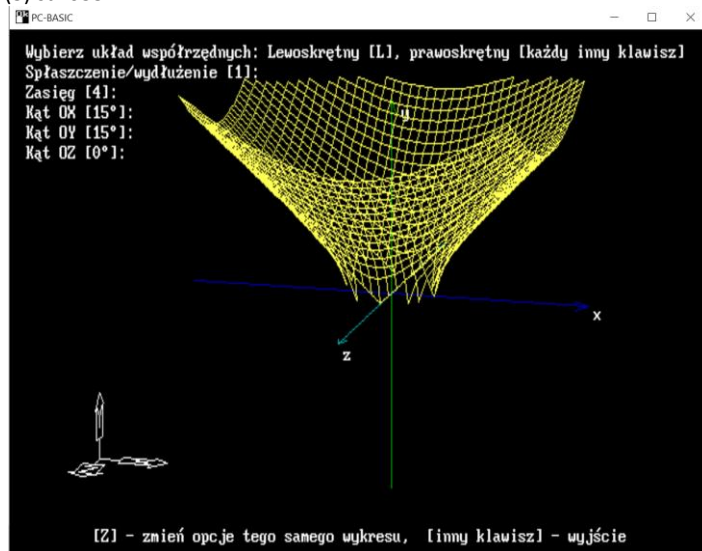
(3) str.636



12990 IF X\*X+Z\*Z-1 < 0 THEN BLAD=1: GOTO 14050

13000 Y = SQR(X\*X+Z\*Z-1)

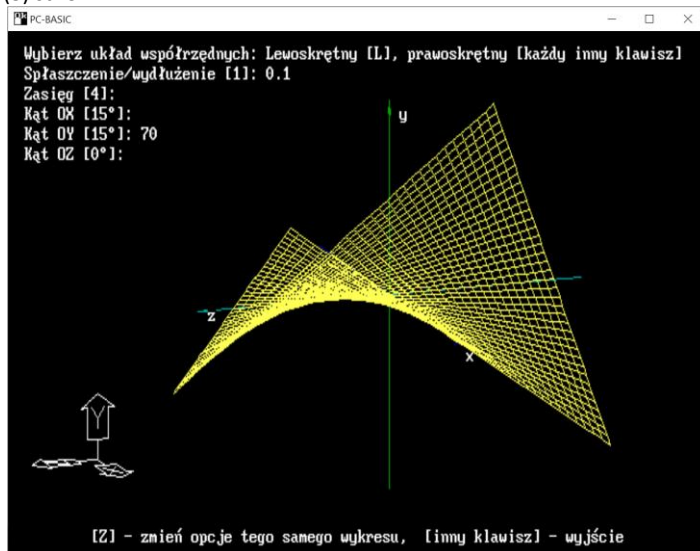
(3) str.635



13000 Y = 2\*X\*Z <-- saddle (siodło)

y = 2xz

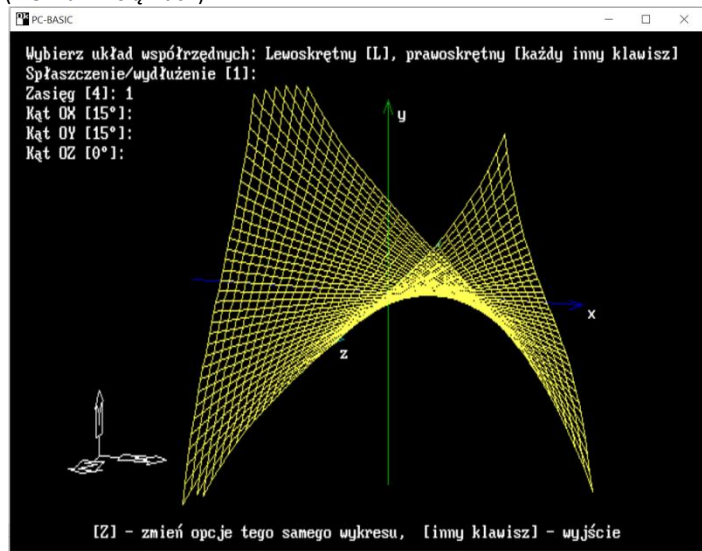
(3) str.S-141



13000 Y = TAN(X\*Z)

y = Tan(xz)

(nie ma w książkach)



- 
- (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

Poniżej jest przykład zastosowania programu lecz nie do wykresu funkcji ale do ogólnie pojętej grafiki przestrzennej.

```
Leszek Buczek - Sprzęgło Oldhana

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) ?

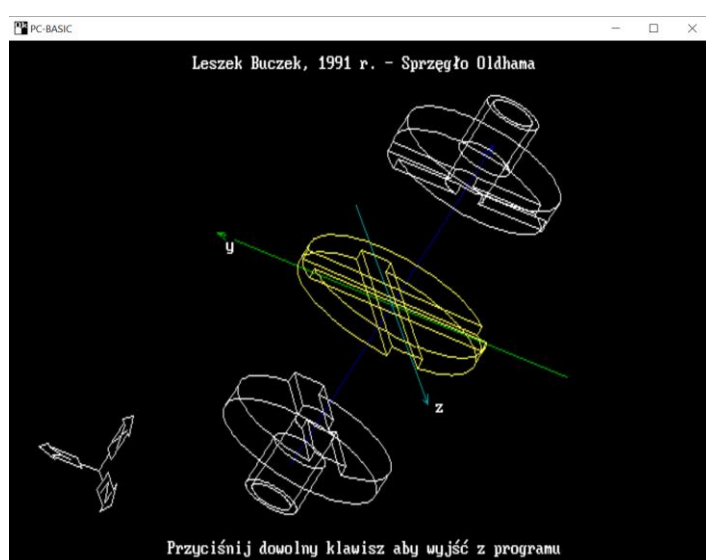
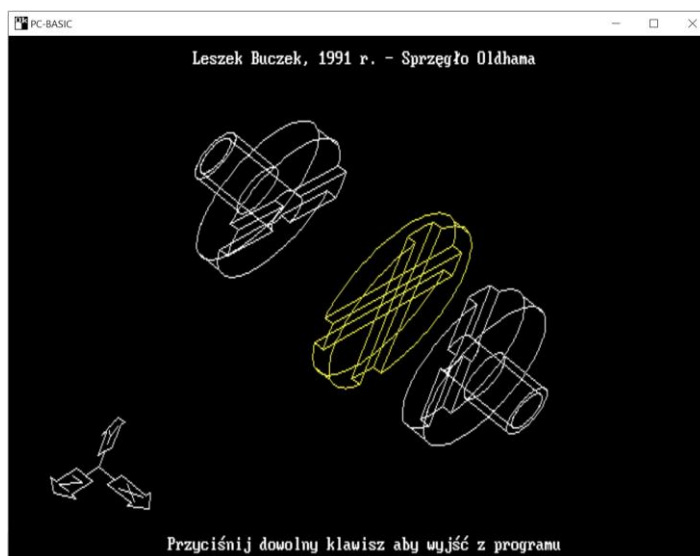
Kąt OX (°) 30
Kąt OY (°) 40
Kąt OZ (°) -20
```

```
Leszek Buczek - Sprzęgło Oldhana

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
```



## Jak to robią inne aplikacje?

Na przykładzie powyżej podanej pierwszej funkcji

' --- Równanie funkcji  $Y=f(X,Z)$  ---

$$V=-(X*X+Z*Z)$$

$$VV=-((X-2)*X+(Z-.5)*Z)$$

$$Y=5*EXP(V)-2*EXP(VV)$$

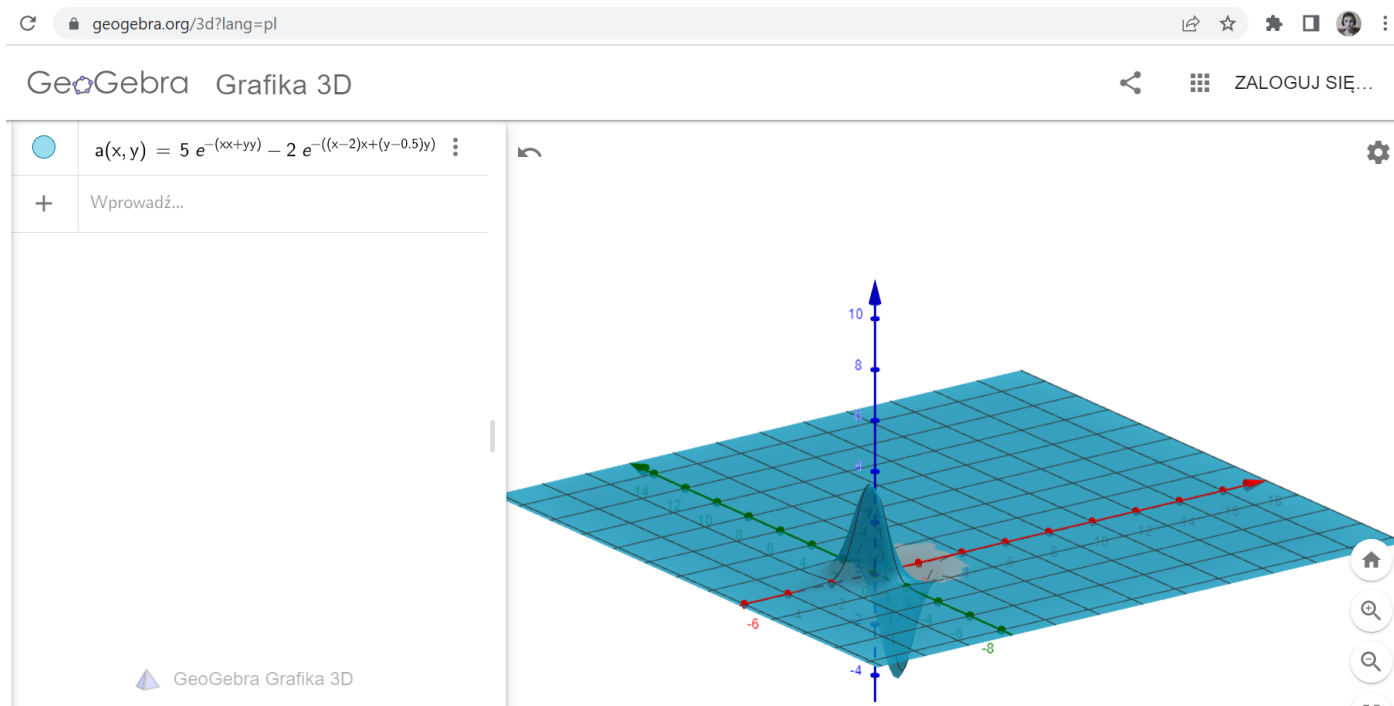
' -----

## Jak to robi GeoGebra

GeoGebra Wykres 3D ← <https://www.geogebra.org/3d?lang=pl>

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





Ogólnie poprawne ale oprócz obrotów nie mogę nic z tym zrobić

## Jak to robi *Python*

Aby otrzymać obraz figury przestrzennej w dowolnym jej ustawieniu trzeba importować takie cudotwórcze mechanizmy jak:

- OpenGL.GL/GLU/GLUT
- vpython

lub coś podobnego.

Jeżeli z nich zrezygnuję to zostaje mi *matplotlib* (poniżej jest też *numpy*, którego nie muszę tu używać). Jakkolwiek wynik jest dobry, ale tylko w jednym ustawieniu - typowo jednostronne zastosowanie.

Poniżej kod do skopiowania do Python'a i wynik:

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

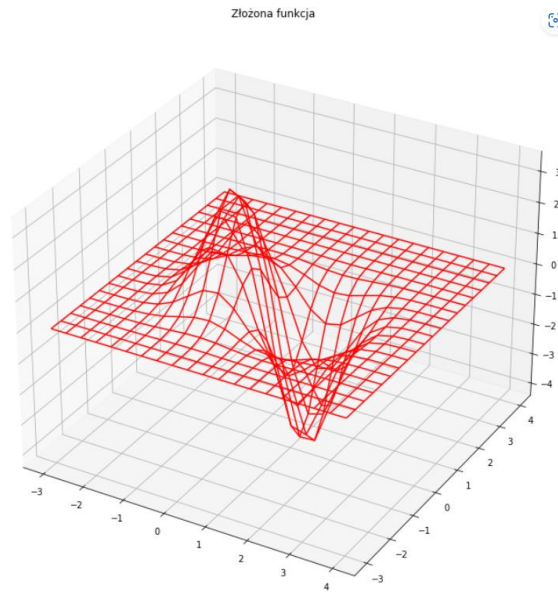
def f(x, y):
    return 5*np.exp(-(x*x+y*y)) - 2*np.exp(-((x-2)*x+(y-0.5)*y))
# V=-(X*X+Z*Z)
# VV=-( (X-2)*X+(Z-.5)*Z)
# Y=5*EXP(V)-2*EXP(VV)

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');
```



Takie rzeczy to od biedy zrobiłby chyba nawet Excel.