

Ćwiczenie 10. Mathematica: Wizualizacja 2D i 3D

10.1. Definiowanie funkcji

Mathematica wykorzystuje dużą liczbę wbudowanych funkcji matematycznych. Jeśli funkcja nie ma standardowego skrótu, to odpowiada jej nazwa z języka angielskiego. W każdym razie nazwa funkcji wbudowanej zaczyna się wielką literą. Jeśli nazwa funkcji składa się z kilku słów, to słowa piszemy jeden po drugim bez spacji, a każde słowo piszemy wielką literą. Argumenty funkcji są zawarte w nawiasach kwadratowych. Przykłady niektórych funkcji matematycznych są wymienione w poniższej tabeli.

Sqrt[x]	Pierwiastek kwadratowy ($\sqrt{x}$)
Exp[x]	funkcja wykładnicza (e^x)
Log[x]	logarytm naturalny ($\ln x$)
Log[b,x]	funkcja logarytmiczna ($\log_b x$)
Sin[x], Cos[x], Tan[x], Cot[x]	funkcje trygonometryczne
ArcSin[x], ArcCos[x], ArcTan[x], ArcCot[x]	funkcje cyklometryczne
Abs[x]	moduł liczby x
Min[lista]	podaje minimalny element listy
Max[lista]	podaje maksymalny element listy

Możemy zdefiniować funkcję stosując palletes (npr. palleta *Basic Math Assistant* klikniemy umieszczamy kursor w miejscu $\sqrt{x}$ i klikamy w tym miejscu)

`f[x_] :=  $\sqrt[3]{x} - 2$`

Funkcja `f[x]` określa funkcję $x^{\frac{1}{3}} - 2$.

Uwaga. Nazwa własna funkcji zaczyna się małą literą. Zmienne (zmienną) funkcji piszemy w nawiasach kwadratowych. Symbol `x_` oznacza parametr formalny (zmienną) funkcji, który zastępuje się przy wykonaniu konkretną zmienną lub wartością. Symbol (bez spacji) „:=” oznacza zdefiniowanie (w naszym przypadku funkcji `f[x]`), przy czym *Mathematica* zapamięta pod funkcją cały podany wzór nie obliczając go.

`f[x]`

$-2 + x^{1/3}$

`f[y]`

$-2 + y^{1/3}$

Znajdziemy wartości tej funkcji przy $x = 27$ i $x = e$

`f[27]`

1

`f[e]`

$-2 + e^{1/3}$

Istnieje możliwość określenia nowych funkcji przez określone wcześniej

`g[x_] := f[x]3`

`g[64]`

8

`In[3]:=`

`f1[x_] :=  $x^3 - 2x^2 + 3x - 2$`

Czyszczenie funkcji

Po zakończeniu pracy z funkcją, pozbyć się jej. Dlaczego? Jednym z powodów jest to, że możesz zapomnieć o funkcji i później w trakcie sesji spróbować użyć jej nazwy do czegoś innego. Ale *Mathematica* nie zapomina, co może prowadzić do wszelkiego

rodzaju nieporozumień. Innym powodem jest to, że pozbywając się funkcji, zwalniasz trochę pamięci, pozostawiając sobie więcej miejsca do pracy. Aby sprawdzić, czy litera lub słowo zostało zdefiniowane jako funkcja, użyj polecenia ? tak jak w przypadku wbudowanego polecenia Mathematica:

In[4]:=

? f1

Out[4]=

Symbol
Global`f1
Definitions
f1 [x_] := x ³ - 2 x ² + 3 x - 2
Full Name Global`f1
^

Oznacza to, że zmienna f jest nadal przechowywana w pamięci. Możemy użyć polecenia Clear, aby ją usunąć, tak jak usuwa się wartość stałej:

In[5]:= **Clear [f1]**

[wyczyść](#)

Jeśli teraz użyjesz polecenia ?, nie znajdziesz takiej definicji:

In[6]:= **? f1**

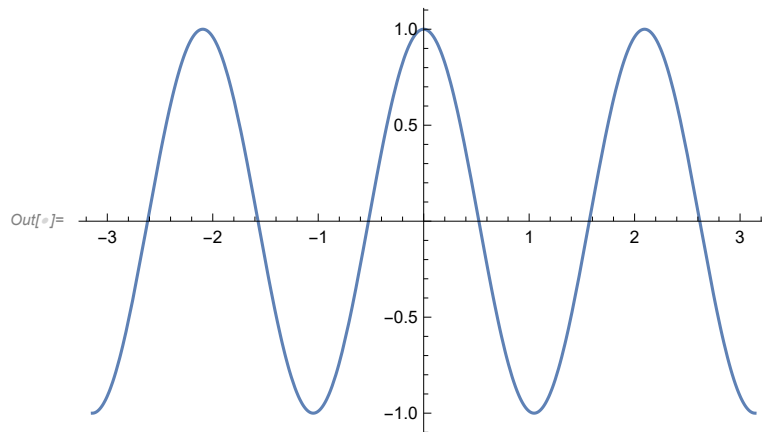
Out[6]=

Symbol
Global`f1
Full Name Global`f1
^

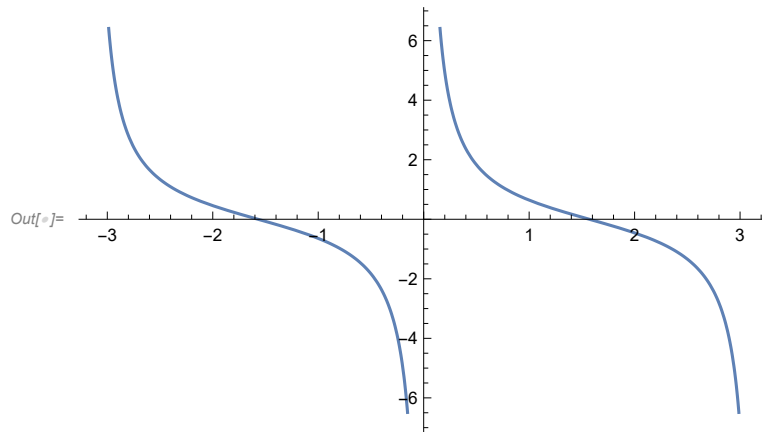
10.2. Wykresy funkcji

Do sporządzania wykresów funkcji jednej zmiennej $y = f(x)$ istnieje polecenie **Plot** format, który ma postać: **Plot**[f[x], {x, x_{min}, x_{max}}], gdzie f[x] funkcja zmiennej x, która należy do przedziału [x_{min}, x_{max}]. Zauważmy, że polecenie **Plot** pozwala rysować wykresy tak funkcji ciągłych jak i mających w przedziale [x_{min}, x_{max}] punkty osobliwe. Np., narysujemy wykresy funkcji Cos[3 x] i Cot[x-π] w przedziale [-π, π].

In[]:=

Plot[Cos[3 x], {x, - π , π }]**W**... **cosinus**

In[]:=

Plot[Cot[x - π], {x, - π , π }]**W**... **cotangens**

Przy rysowaniu wykresu *Mathematica* musi wyznaczyć punkty przedziału $[x_{\min}, x_{\max}]$ w których należy obliczyć wartości funkcji, wyznaczyć obszar zmiany funkcji dla wyobrażenia na wykresie, jaką wybrać skalę, jak zbudować osie współrzędnych i tak dalej. Zarządzanie tymi procesami odbywa się za pomocą kilku opcji, z których każda ma określoną nazwę, a odpowiadająca wartość opcji jest określona przez polecenie: Opcja $\rightarrow$ Value. Listę opcji możemy znaleźć za pomocą polecenia:

In[]:=

Options[Plot]**opcje** **wykres**

Out[]:=

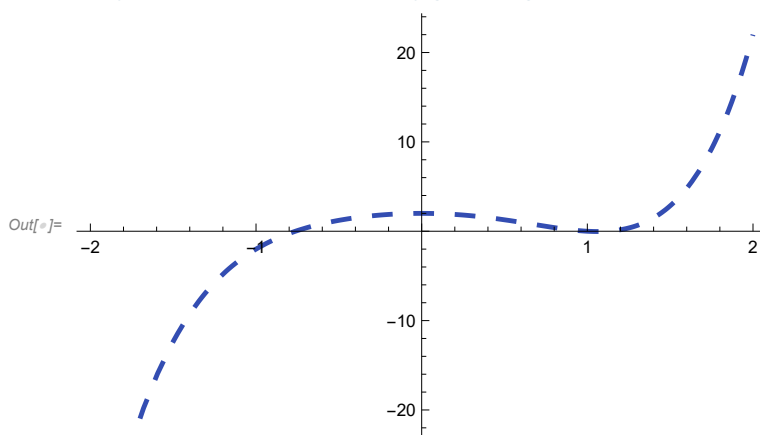
```
{AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , Axes → True, AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {},
Background → None, BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic,
ColorFunctionScaling → True, ColorOutput → Automatic, ContentSelectable → Automatic, CoordinatesToolOptions → Automatic,
DisplayFunction → $DisplayFunction, Epilog → {}, Evaluated → Automatic, EvaluationMonitor → None, Exclusions → Automatic,
ExclusionsStyle → None, Filling → None, FillingStyle → Automatic, FormatType → TraditionalForm, Frame → False,
FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {}, GridLines → None,
GridLinesStyle → {}, ImageMargins → 0., ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic,
LabelingSize → Automatic, LabelStyle → {}, MaxRecursion → Automatic, Mesh → None, MeshFunctions → {#1 &},
MeshShading → None, MeshStyle → Automatic, Method → Automatic, PerformanceGoal → $PerformanceGoal, PlotLabel → None,
PlotLabels → None, PlotLayout → Automatic, PlotLegends → None, PlotPoints → Automatic, PlotRange → {Full, Automatic},
PlotRangeClipping → True, PlotRangePadding → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
PlotTheme → $PlotTheme, PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &), RotateLabel → True,
ScalingFunctions → None, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}, WorkingPrecision → MachinePrecision}
```

Dlatego, żeby zmienić wygląd wykresu, na przykład ponownie wprowadzić osie współrzędnych i nowy kolor linii, trzeba dodać do podstawowego polecenia **Plot** trzecie i następne argumenty: **Plot**[*f*[*x*], {*x*, *xmin*, *xmax*}, *option*→*value*]. Następnie rozpatrzmy niektóre z tych opcji.

Opcja **PlotStyle** daje możliwość zmiany grubości, koloru i stylu krzywej na wykresie. Żeby zmienić kilka parametrów krzywej, dyrektywy odpowiadające należy określić w postaci listy. Grubość linii lub krzywej określa się opcją **Thickness**, argument której jest stosunkiem grubości linii do szerokości całego wykresu. Aby określić kolor linii używany kilku dyrektyw. Dyrektywa **GrayLevel**, argument której może mieć wartości od 0 do 1, określa różne odcienie szarości (kolor czarny odpowiada wartości **GrayLevel** [0], kolor biały – **GrayLevel** [1]). Dyrektywa **RGBColor** [*r*, *g*, *b*] określa kolor linii jako połączenie kolorów czerwonego, zielonego i niebieskiego. Intensywność każdego z tych trzech składników jest określona przez wartość argumentu *r*, *g* i *b*, który zmienia się 0 do 1. Różne odcienie kolorów mogą być również określone przez dyrektywę **Hue**[*h*], gdzie parametr *h* zmienia się od 0 do 1. Dyrektywa **Dashing**[{*d*₁, *d*₂, ...}] pozwala narysować krzywą przerywaną. Np.:

In[]:=

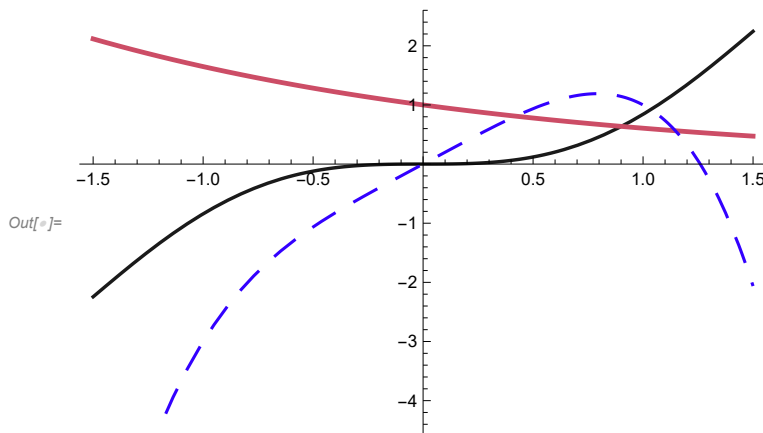
```
wyk1 = Plot[x^5 - 3 x^2 + 2, {x, -2, 2}, PlotStyle → {Thickness[0.007], RGBColor[0.2, 0.3, 0.7], Dashing[{0.03}]}
```

wykres**styl grafiki****grubość****kolor RGB****specyfikacja linii prz**

Przy zbudowaniu na jednej płaszczyźnie kilka krzywych, możemy określić własny styl dla każdej z nich. W tym przypadku opcja **PlotStyle** będzie listą stylów w takiej kolejności, w jakiej są zapisane w pierwszym argumentie polecenia **Plot**.

In[]:=

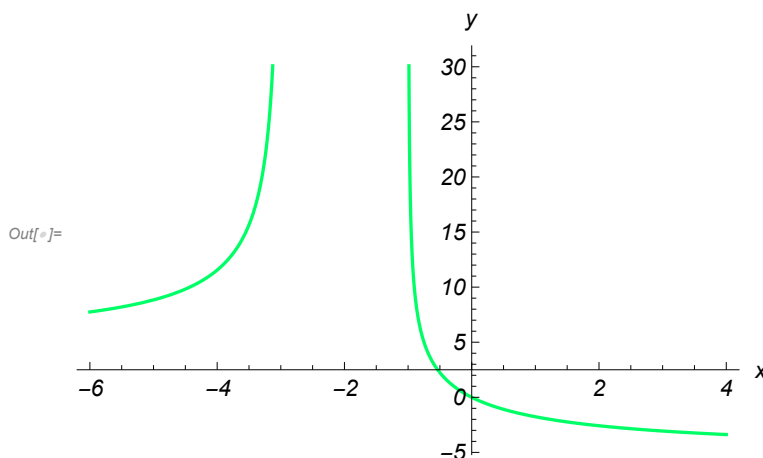
```
wyk2 = Plot[{x^2 Sin[x], Exp[-x/2], 2 x - x^4}, {x, -1.5, 1.5}, PlotStyle →
  {GrayLevel[0.1], Thickness[0.005]}, {Thickness[0.007], RGBColor[0.8, 0.3, 0.4]}, {Hue[0.7], Dashing[{0.04, 0.03]}}]
```



Kilka opcji określa typ osi współrzędnych na wykresie. Opcja **Axes→True** wskazuje na niezbędność rysowania osi, w przeciwnym przypadku wybieramy opcję **Axes→False**. Opcja **AxesStyle** określa styl osi i może przyjmować takie same wartości jak i opcja **PlotStyle**. Aby określić położenie początku osi współrzędnych w opcji **AxesOrigin**, trzeba wyznaczyć współrzędne tego punktu przecięcia. Dla każdej osi możemy wprowadzić nazwę za pomocą opcji **AxesLabel**. Np.:

In[]:=

```
wyk3 = Plot[-5 x/Sqrt[(x + 2)^2 - 1], {x, -6, 4}, AxesStyle → Directive[Black, 12, Italic],
  AxesOrigin → {0, 2.5}, AxesLabel → {"x", "y"}, PlotStyle → {Thickness[0.005], Hue[0.4]}]
```

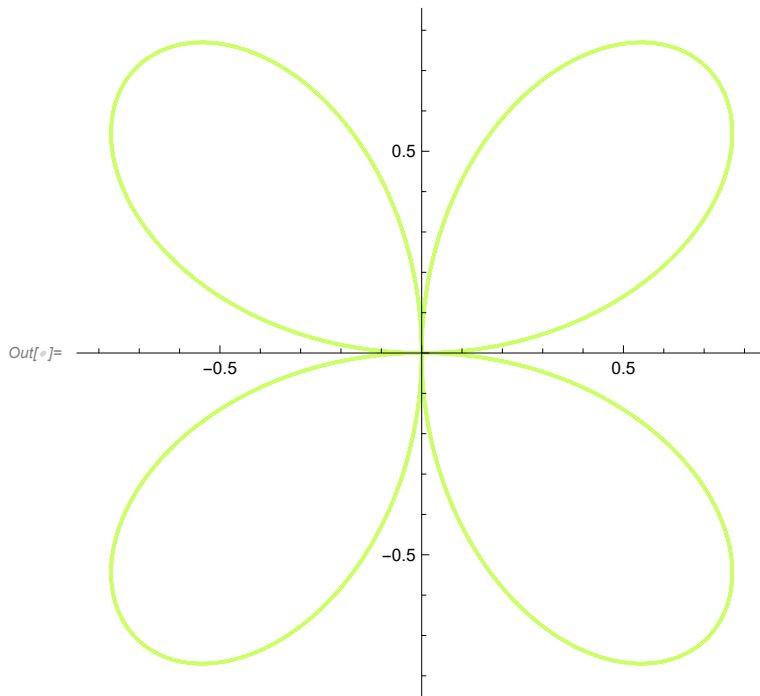


Dla sporządzania dwuwymiarowego wykresu funkcji zadanej parametrycznie używamy instrukcji **ParametricPlot**, format której ma postać: **ParametricPlot[{x[t], y[t]}, {t, tmin, tmax}, option→value]**. Ta instrukcja ma te same opcje co i instrukcja **Plot**. Pozwala ona budować tak wykres jednej jak i wykresy kilku funkcji.

In[]:=

```
ParametricPlot[Cos[3 t] Sin[6 t], Sin[3 t] Sin[6 t]], {t, 0, 2 π}, PlotStyle → {Thickness → 0.005, RGBColor[0.8, 1, 0.4]}]
```

[wykres paramet...] [cosinus] [sinus] [sinus] [styl grafiki] [grubość] [kolor RGB]



Tak więc, jeśli funkcja jednej zmiennej jest określona analitycznie, to aby ją zwizualizować, możemy użyć polecenia **Plot** i **ParametricPlot**. Jednak niekiedy istnieje sytuacja, gdy wzór funkcji nie jest znany, a wiadomy jest ciąg liczbowych wartości funkcji postaci $\{y_1, y_2, \dots, y_n\}$ lub $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. Dla wizualizacji takich zespołów (danych liczbowych) wykorzystujemy instrukcję **ListPlot**:

ListPlot $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_n, y_n\}\}$ wykreśla punkty $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

Określimy, na przykład, listę współrzędnych punktów $(i, 3 \cos i)$, gdzie i zmienia się od 0 do 3π z krokiem $\frac{\pi}{6}$ i przedstawimy graficznie te punkty za pomocą instrukcji **ListPlot**.

In[]:=

```
dat = Table[{i, 3 Cos[i]}, {i, 0, 3 π, π/6}]
```

[tabela] [cosinus]

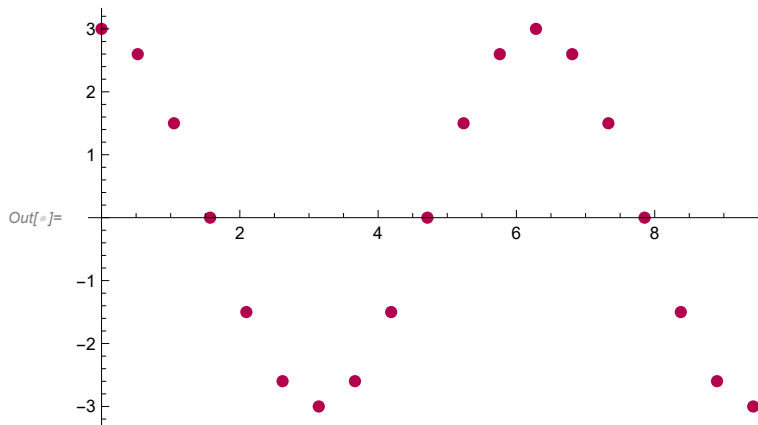
Out[]:= $\left\{ \{0, 3\}, \left\{\frac{\pi}{6}, \frac{3\sqrt{3}}{2}\right\}, \left\{\frac{\pi}{3}, \frac{3}{2}\right\}, \left\{\frac{\pi}{2}, 0\right\}, \left\{\frac{2\pi}{3}, -\frac{3}{2}\right\}, \left\{\frac{5\pi}{6}, -\frac{3\sqrt{3}}{2}\right\}, \{\pi, -3\}, \right.$
 $\left. \left\{\frac{7\pi}{6}, -\frac{3\sqrt{3}}{2}\right\}, \left\{\frac{4\pi}{3}, -\frac{3}{2}\right\}, \left\{\frac{3\pi}{2}, 0\right\}, \left\{\frac{5\pi}{3}, \frac{3}{2}\right\}, \left\{\frac{11\pi}{6}, \frac{3\sqrt{3}}{2}\right\}, \{2\pi, 3\}, \right.$
 $\left. \left\{\frac{13\pi}{6}, \frac{3\sqrt{3}}{2}\right\}, \left\{\frac{7\pi}{3}, \frac{3}{2}\right\}, \left\{\frac{5\pi}{2}, 0\right\}, \left\{\frac{8\pi}{3}, -\frac{3}{2}\right\}, \left\{\frac{17\pi}{6}, -\frac{3\sqrt{3}}{2}\right\}, \{3\pi, -3\} \right\}$

Dla określenia rozmiaru i koloru punktów używamy opcji **PlotStyle** i dyrektywy **PointSize**.

In[]:=

```
ListPlot[dat, PlotStyle -> {RGBColor[0.7, 0, 0.3], PointSize[0.018]}]
```

[wykres danych](#) [styl grafiki](#) [kolor RGB](#) [rozmiar kropki](#)

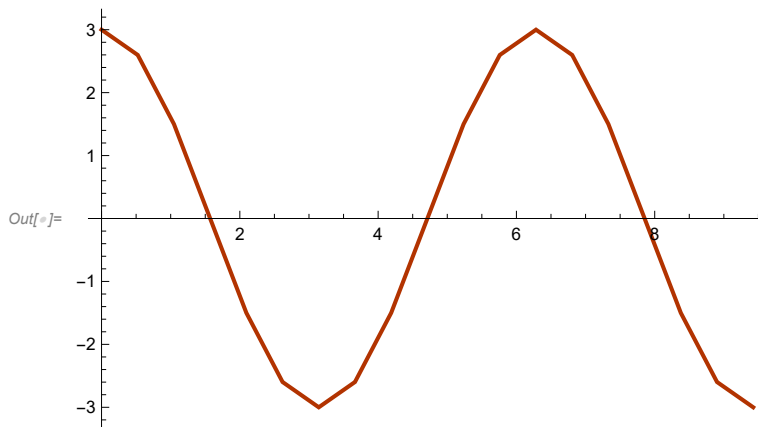


Punkty na wykresie możemy połączyć między sobą odcinkami prostej za pomocą instrukcji **ListLinePlot**. Dla opcji **PlotStyle** zamiast dyrektywy **PointSize** należy określić grubość linii **Thickness**.

In[]:=

```
ListLinePlot[dat, PlotStyle -> {RGBColor[0.7, 0.2, 0], Thickness[0.006]}]
```

[wykres liniowy](#) [styl grafiki](#) [kolor RGB](#) [grubość](#)



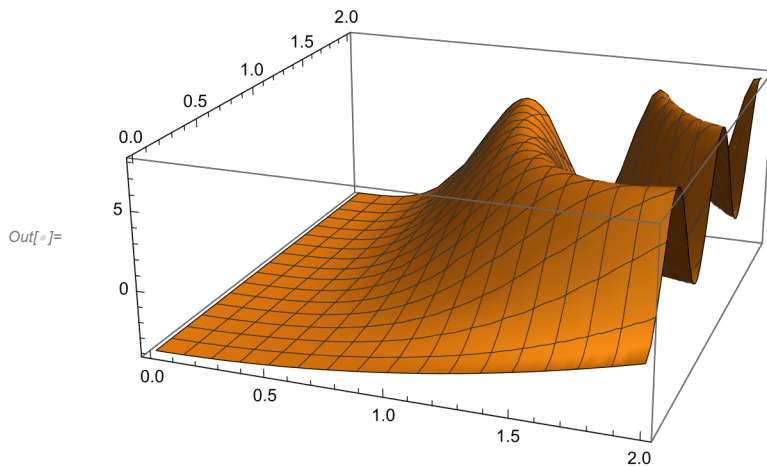
Pozostałe opcje instrukcji **ListPlot** są podobne do odpowiednich opcji instrukcji **Plot**.

Funkcją dwóch zmiennych $f(x, y)$ określa się powierzchnię w przestrzeni trójwymiarowej. Wizualizację 3D tej powierzchni możemy utworzyć przy pomocy instrukcji **Plot3D**, format której ma postać: **Plot3D[f[x,y], {x, xmin, xmax}, {y, ymin, ymax}]**, gdzie $f[x, y]$ jest określoną funkcją zmiennych x i y , które należą odpowiednio do przedziałów $[xmin, xmax]$ i $[ymin, ymax]$. Np. $f(x, y) = x^2 - 4 \cos(x^2 y)$, $x \in [0, \pi]$, $y \in [0, \pi]$.

In[]:=

```
Plot3D[x^2 - 4 Cos[2 x^2 y], {x, 0, 2}, {y, 0, 2}]
```

[wykres trój...](#) [cosinus](#)



Przy wywołaniu instrukcji **Plot3D** możemy dopisać różne opcje. Niektóre opcje (**AspectRatio**, **Axes**, **AxesLabel**, **AxesStyle**, **Background**, **BaseStyle**, **Epilog**, **ImageSize**, **PlotLabel**, **PlotPoints**, **PlotRange**, **PlotRegion**, **Prolog**, **Ticks**) są podobne do odpowiednich opcji przy 2D wizualizacji.

Jeśli instrukcja **Plot3D** zawiera tylko trzy argumenty, to przy zbudowaniu powierzchni wykorzystujemy wartości opcji ustalone domyślnie, żeby otrzymać listę opcję razem z określonymi wartościami stosujemy polecenie **Options[Plot3D]**. Dlatego, aby zmienić wartości którekolwiek opcji, wystarczy dołączyć w jakości czwarty i następne argumenty instrukcji **Plot3D** w dowolnej kolejności, wybierając dla nich odpowiednie wartości: **Plot3D**[$f[x, y]$, { x , x_{min} , x_{max} }, { y , y_{min} , y_{max} }, *option* → *value*].

Rozpatrzmy tylko niektóre z tych opcji. Dla większej dokładności wyobrażenia powierzchni możemy zwiększyć wartość domyślną opcji **PlotPoints**, przy tym ilość punktów obliczania może być różna względem osie x i y , np., **PlotPoints** → {50, 40}. Dla ułatwienia wyobrażenia przestrzennej struktury powierzchni domyślnie nanosi się na je kratkę prostokątną (opcja **Mesh** → **True**), a wykres znajduje się wewnątrz ramy równoległoscianu prostokątnego (opcja **Boxed** → **True**). Style linii kratki i ramy są określone odpowiednio za pomocą opcji **MeshStyle** i **BoxStyle**. Stosunek długości boków ramy zadaje się opcją **BoxRatios**. Dla każdej ściany ramy możemy nanieść kratkę za pomocą polecenia **FaceGrids** → **All**.

Dla wizualizacji 3D funkcji jednej lub dwóch zmiennych stosujemy instrukcję **ParametricPlot3D**:

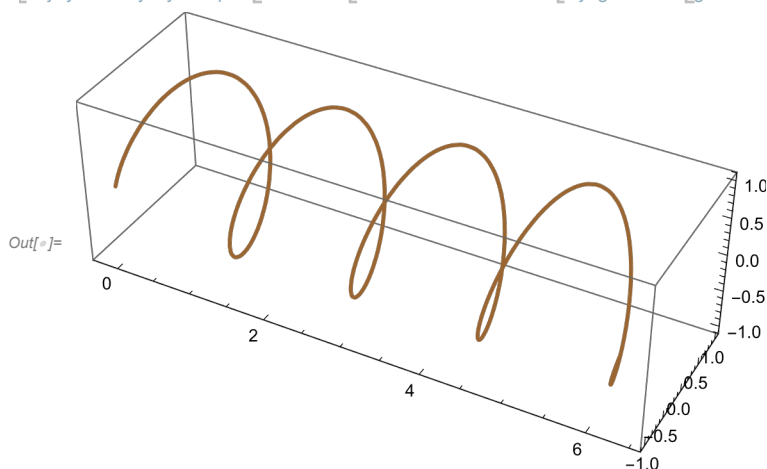
dla krzywej – **ParametricPlot3D**[{ $f_x[t]$, $f_y[t]$, $f_z[t]$ }, { t , t_{min} , t_{max} }, *option* → *value*],

dla powierzchni – **ParametricPlot3D**[{ $f_x[u, v]$, $f_y[u, v]$, $f_z[u, v]$ }, { u , u_{min} , u_{max} }, { v , v_{min} , v_{max} }, *option* → *value*].

In[]:=

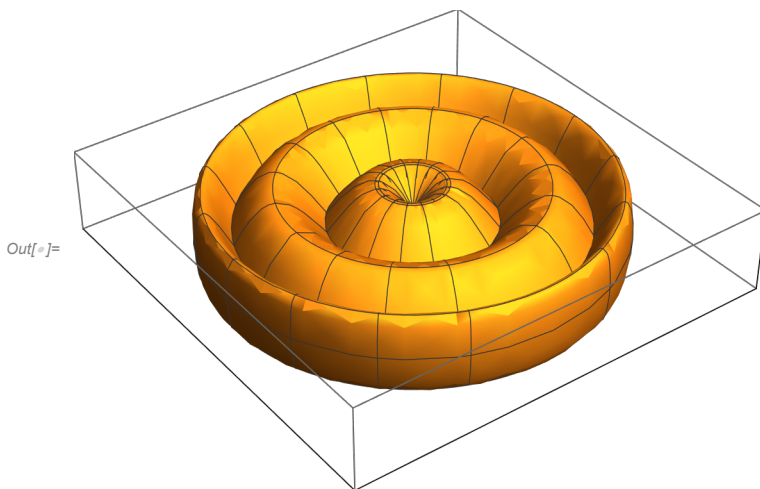
```
ParametricPlot3D[{u, -Cos[4 u], Sin[4 u]}, {u, 0, 2 Pi}, PlotStyle -> {Thickness[0.006], Brown}]
```

[trójwymiarowy wykres p...](#) [cosinus](#) [sinus](#) [styl grafiki](#) [grubość](#) [brązowy](#)




```
in[ ]:=
ParametricPlot3D[{(2 - 3 Sin[θ]) Cos[φ], (2 - 3 Sin[θ]) Sin[φ], Sin[4 θ]},

```



Uwaga. Klikając prawym przyciskiem myszki na wykresie i przesuwając ją możemy dowolnie nim obracać. To oznacza, że możemy interaktywnie animować grafikę trójwymiarową za pomocą myszki.

Zadania

Zadanie 1. Narysuj wykres funkcji od -10 do 10:

- a) $\sin(\pi - \cos x)$;
b) $\operatorname{tg}(1.5 - \cos x)$;
c) $\sin^2(\pi + 2 \cos(2x))$;
d) $\sinh(1 - e^{\sin x})$.

Zadanie 2. Narysować wykres funkcji $f(x) = x^3 + x + 2$ i znaleźć pierwiastek równania $f(x) = 0$. Narysować punkt, odpowiadający pierwiastkowi na wykresie funkcji.

Zadanie 3. Narysować wykresy funkcji $f(x) = \sin(2x)$ i jej stycznej w punkcie $x = \frac{\pi}{4}$ na przedziale $[-\pi; \pi]$.

Zadanie 4. Narysować wykres funkcji $f(x) = 2x^3 - 5x^2 + 4x + 1$ i znaleźć punktów ekstremów. Zrobić wizualizację na przedziale $[0.5; 1.2]$.

Zadanie 5. Narysować wykresy trzech funkcje BesselJ[n,x] ($n=1,2,3$) różnej grubości i pokolorowanych odpowiednio w kolor czerwony, żółty, zielony.

Zadanie 6. Narysować wykres funkcji:

- $z = F(x, y) = xy^2$, gdzie $-2 \leq x \leq 2$, $-3 \leq y \leq 3$;
- $z = F(x, y) = x^3 - y^3 - 1$, gdzie $-2.5 \leq x \leq 2.5$, $-2 \leq y \leq 2$;
- $F(t) = \{\cos t, \sin t, t/2\}$, gdzie $-2\pi \leq t \leq 2\pi$;
- $F(u, v) = \{\cos u, \sin v, \sin u \sin v, \cos v\}$, gdzie $-2\pi \leq u \leq 2\pi$, $-\pi \leq v \leq \pi$;
- $F(u, v) = \{\cos 4 + (3 + \cos v) \sin u, 4 + (3 + \cos v) \cos u, 4 + \sin v\}$, gdzie $0 \leq u \leq 2\pi$, $0 \leq v \leq \pi$.

Zadanie 7. Narysować wykres funkcji:

- a) $\sin(2x) + \cos(3y) = 1/2$, gdzie $-\pi/6 \leq x \leq 3\pi - \pi/6$, $-\pi/3 \leq y \leq 3\pi$;
b) $3x^4 = y^2 + z^2$, gdzie $-2 \leq x \leq 2$, $-2 \leq y \leq 2$, $-2 \leq z \leq 2$.
-