

# 第 1 章 グラフの作成

本書では、次の 2 つの方法を紹介する.

1. PGFPlots による方法
2. GNUPlot による方法

なお、これらの他にも TikZ の datavisualization ライブラリによる方法もある. また、多くの数学ソフトウェアは  $\text{\LaTeX 2}_\epsilon$  に対応した出力、あるいは PDF 出力を備えている.

## 1.1 PGFPlots によるグラフ作成

```
\begin{tikzpicture}
  \begin{axis}[width=7cm]
    \addplot [smooth,samples=100,domain=-pi:pi] {cos(deg(x))};
    \addlegendentry{$\cos(x)$};
  \end{axis}
\end{tikzpicture}
```

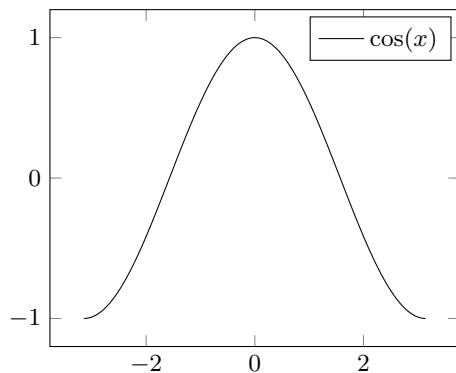


図 1.1: PGFPlots による  $\cos(x)$  のグラフ

## 1.2 GNUPlot によるグラフ作成

あああ<sup>1)</sup>.

---

1) `set term tikz` としている資料もあるが、これは `set terminal lua tikz` の省略形である.

```
gnuplot> set terminal lua tikz size 7cm,5cm
gnuplot> set output "gnuplot.tex"
gnuplot> set xrange [-pi:pi]
gnuplot> plot cos(x) linetype rgb "black" title '$\cos(x)$'
gnuplot> exit
```

`\usepackage{gnuplot-lua-tikz}` を追加

`\input{gnuplot.tex}`

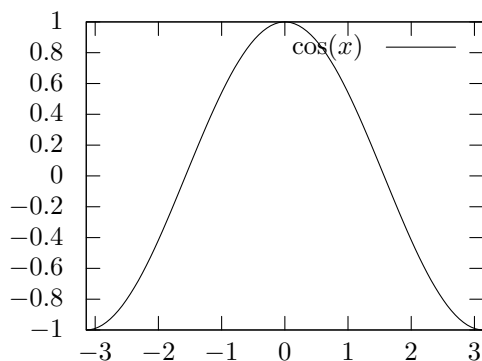


図 1.2: GNUPlot による  $\cos(x)$  のグラフ

## 1.3 その他のグラフ

3D グラフなども同様の要領で作成できる．以下に PGFPlots を利用した例を示す．

```
\begin{tikzpicture}
  \begin{axis}[width=7cm,colormap/blackwhite]
    \addplot3 [surf,miter limit=1,samples=30,domain=-2:2] {exp(-(x^2+y^2))};
  \end{axis}
\end{tikzpicture}
```

## 1.4 補遺

### 1.4.1 グラフ描画の精度

通常, PGFPlots は  $\text{T}_\text{E}\text{X}$  上で実装された浮動小数点数演算を利用して, 関数のグラフを計算する．しかし, 処理系が  $\text{Lua}\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  であり, `compat` に 1.12 以上のバージョンが指定されていれば,

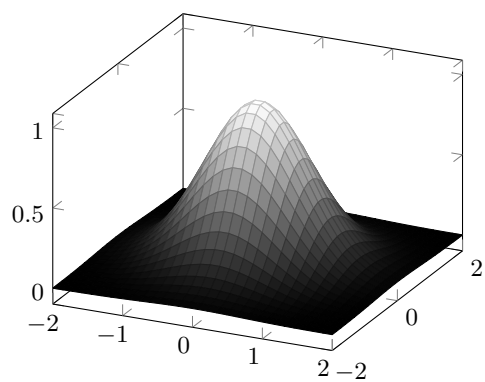


図 1.3:  $e^{-(x^2+y^2)}$  のグラフ

PGFPlots は Lua の数学エンジンを利用して計算を行う．これは  $\text{T}_{\text{E}}\text{X}$  上の実装より高速かつ正確である [?, p.54].