第 5 章　数学公式与特殊符号

与 Word 相比，LaTeX 的一大优势就是数学公式的排版比较优美。在 Word 中，需要用户自己调节公式的大小和位置，在 LaTeX 中，这些都可以通过命令行的形式控制。

5.1　数学符号的基本显示

LaTeX 使用一种特殊的模式来排版数学符号和公式(mathematics)。段落中的数学表达式应该置于 \和\、$ 和 $ 或者\begin{math}和\end{math}之间。对于比较短的数学公式或者符号，使用 $ 和 $ 即可，如图 5.1 所示。对应 PDF 中的数学符号如图 5.2 所示。

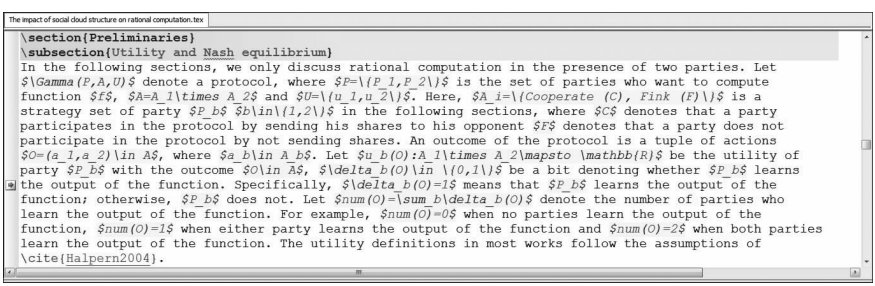
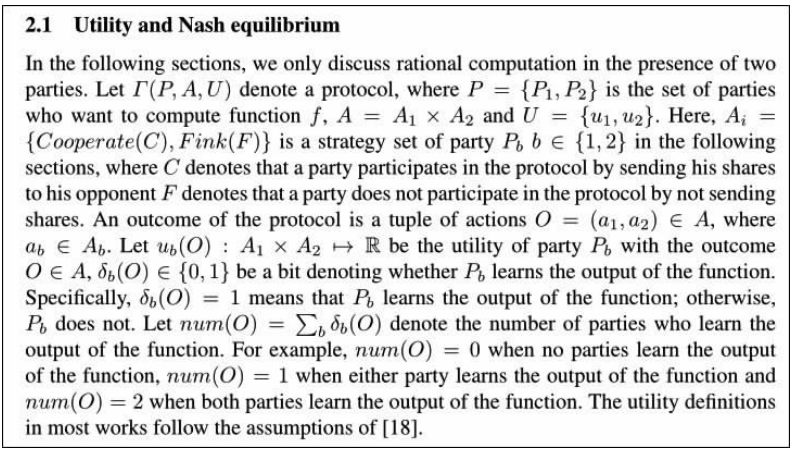


图 5.1　简单数学符号命令行

图 5.1 中都是简单的数学符号，有些数学符号可以直接输入，例如 A、P 等，有些数学符号有特殊的命令对应，例如 Γ、δ 等，它们对应的命令为\Gamma、\delta。

在数学符号中常见的就是上标和下标。下标用符号“_”表示，上标用符号“^”表示。需要注意的是，如果下标符号不止一个，就需要将所有的下标用{}括起来。例如，A_{sum} 所对应的命令为 A_{sum}而不是 A_sum。如果输入 A_sum，所产生的数学符号为 A_sum。用户需要格外注意这个问题，因为出现这类错误，LaTeX 是不会报错的，因为 LaTeX 认



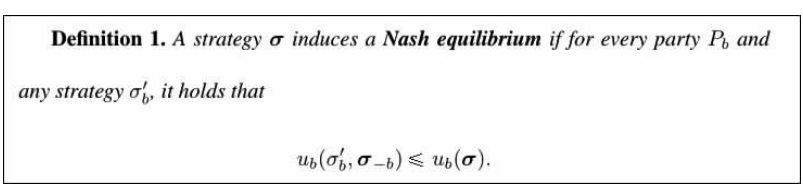
2.1 Utility and Nash equilibrium

In the following sections, we only discuss rational computation in the presence of two parties. Let $\Gamma(P, A, U)$ denote a protocol, where $P = \{P_1, P_2\}$ is the set of parties who want to compute function f, $A = A_1 \times A_2$ and $U = \{u_1, u_2\}$. Here, $A_i = \{Cooperate(C), Fink(F)\}$ is a strategy set of party P_b $b \in \{1, 2\}$ in the following sections, where C denotes that a party participates in the protocol by sending his shares to his opponent F denotes that a party does not participate in the protocol by not sending shares. An outcome of the protocol is a tuple of actions $O = (a_1, a_2) \in A$, where $a_b \in A_b$. Let $u_b(O) : A_1 \times A_2 \mapsto \mathbb{R}$ be the utility of party P_b with the outcome $O \in A$, $\delta_b(O) \in \{0, 1\}$ be a bit denoting whether P_b learns the output of the function. Specifically, $\delta_b(O) = 1$ means that P_b learns the output of the function; otherwise, P_b does not. Let $num(O) = \sum_b \delta_b(O)$ denote the number of parties who learn the output of the function. For example, $num(O) = 0$ when no parties learn the output of the function, $num(O) = 1$ when either party learns the output of the function and $num(O) = 2$ when both parties learn the output of the function. The utility definitions in most works follow the assumptions of [18].

图 5.2　简单数学符号效果

为这没有语法错误。

对于较大的数学式子，最好的方法是使用显示样式来排版：将它们放置于 \[和 \]、$$ 和 $$ 或\begin{displaymath}和\end{displaymath}之间，这样排版出的公式是没有编号的，如图 5.3 所示。图中的公式 $u_b(\sigma_b', \sigma_{-b}) \leqslant u_b(\sigma)$没有编号。



Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ_b', *it holds that*

$$u_b(\sigma_b', \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}).$$

图 5.3　没有编号的公式

对于没有编号的公式，如果希望在后面引用该公式时，会带来麻烦。最好的情形是，给公式一个编号，这样在后面正文中如果需要再次用到该公式，可以直接提及该公式的编号。

如果希望公式有编号，可以使用 equation 环境来自动为每一个\begin{equation}和\end{equation}之间的公式编号。所对应的命令行如图 5.4 所示，产生的 PDF 文件中的公式如图 5.5 所示。图 5.5 与图 5.3 相比，只是公式 $u_b(\sigma_b', \sigma_{-b}) \leqslant u_b(\sigma)$多了一个编号

(1)。在后面的正文中,如果希望再次引用该公式,就可以写“公式(1)”。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a \textbf{Nash equilibrium} if for every
party $P_b$ and any strategy $\sigma'_b$, it holds that
\begin{equation}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}).
\end{equation}}
```

图 5.4　公式编号的命令行

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}). \tag{1}$$

图 5.5　公式编号的效果图

在 LaTeX 命令行中,如果想在正文中引用带有编号的公式,不需要用户记住每个公式的详细编号,只需要知道每个公式的 label 即可。这一点与图形和表格的引用类似。例如,在图 5.6 中,公式 $u_b(\sigma'_b,\sigma_{-b})\leqslant u_b(\sigma)$的 label 是 def1,用\label{def1}表示,在后面的文章中引用时,需要使用\ref{def1}引用,对应的命令行和所产生的效果如图 5.6 和 5.7 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}).
\end{equation}}

In equation \ref{def1}, if $u_b(\sigma'_b,\vec{\sigma}_{-b})< u_b(\vec{\sigma})$ suffices, the strategy
$\vec{\sigma}$ induces a \textbf{ strict Nash equilibrium}. A (strict) Nash equilibrium guarantees that
no one gains any advantages by deviating from the equilibrium as long as other parties follow it. The
function of (strict) Nash equilibrium is to guarantee parties to follow the protocol.
```

图 5.6　公式标签的标示和引用

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}). \tag{1}$$

In equation 1 if $u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) < u_b(\boldsymbol{\sigma})$ suffices, the strategy $\boldsymbol{\sigma}$ induces a **strict Nash equilibrium**. A (strict) Nash equilibrium guarantees that no one gains any advantages

图 5.7　公式引用后的效果

5.2 多行数学符号显示

对于一些复杂的数学公式，如果长度超过一行，可以考虑使用分行显示。可以使用 aligned 环境，将打算分行显示的公式输入在\begin{aligned}和\end{aligned}之间，如图 5.8 所示。在图 5.8 中，\\是换行符，& 是对齐符，想让公式在哪里对齐，就把 & 放到哪里。所产生的效果如图 5.9 所示。

```
\begin{displaymath}
 \begin{aligned}
&NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
&NU=\rho_1+\frac{\rho_3}{2} \\
&NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
&NU^{--}=-\rho_1+\rho_3
\end{aligned}
\end{displaymath}
```

图 5.8 使用 aligned 命令行

$$
\begin{aligned}
NU^+ &= \rho_1 + \rho_2 + \frac{\rho_3}{3} \\
NU &= \rho_1 + \frac{\rho_3}{2} \\
NU^- &= -\rho_1 + \rho_2 + \frac{\rho_3}{2} \\
NU^{--} &= -\rho_1 + \rho_3
\end{aligned}
$$

图 5.9 使用 aligned 分行后的效果

公式分行还可以使用 array 环境来排版数组(arrays)。它有些类似于 tabular 环境，使用\\命令来分行，如图 5.10 所示。

图 5.8 与图 5.10 的区别如下。

(1) 前者需要使用 & 符号来表示对齐位置，而后者不需要，因为后者类似于一个表格。注意，既然类似于表格，那么需要指定每一列的对齐方式，对齐方式的指定和表格相同，在图 5.10 中，使用 c，表明对齐方式是居中对齐，效果如图 5.11 所示。

```
\begin{displaymath}
 \begin{array}{c}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2} \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
NU^{--}=-\rho_1+\rho_3
\end{array}
\end{displaymath}
```

图 5.10 使用 array 命令行

$$
\begin{array}{c}
NU^+ = \rho_1 + \rho_2 + \frac{\rho_3}{3} \\
NU = \rho_1 + \frac{\rho_3}{2} \\
NU^- = -\rho_1 + \rho_2 + \frac{\rho_3}{2} \\
NU^{--} = -\rho_1 + \rho_3
\end{array}
$$

图 5.11 使用 array 分行后的效果

(2) 前者无须指定列数,后者需要指定列数,并指定每一列的对齐方式。

使用 array 的另一个好处是对于数组或者矩阵的表示。例如,输入一个 3 行 3 列的矩阵,可以使用图 5.12 所示的命令行,通过命令行可以看出,这里每行每列数据的输入和表格类似。\left 和\right 分别表示左大括号和右大括号。生成 PDF 的效果如图 5.13 所示。

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
\rho_1&\rho_2&\frac{\rho_3}{3} \\
\rho_1&0&\frac{\rho_3}{2} \\
\rho_1&\rho_2&\frac{\rho_3}{2}\\
0&0&-\rho_1+\rho_3
\end{array} \right)
\end{displaymath}
```

图 5.12　矩阵对应命令行

$$X = \begin{pmatrix} \rho_1 & \rho_2 & \frac{\rho_3}{3} \\ \rho_1 & 0 & \frac{\rho_3}{2} \\ \rho_1 & \rho_2 & \frac{\rho_3}{2} \\ 0 & 0 & -\rho_1 + \rho_3 \end{pmatrix}$$

图 5.13　矩阵效果

像在 tabular 环境中一样,也可以在 array 环境中画线。例如,分隔矩阵中的元素命令行如图 5.14 所示,效果如图 5.15 所示。但是在数学公式中一般不习惯这样用。

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{c|c|c}
\rho_1&\rho_2&\frac{\rho_3}{3} \\
\hline
\rho_1&0&\frac{\rho_3}{2} \\
\hline
\rho_1&\rho_2&\frac{\rho_3}{2}\\
\hline
0&0&-\rho_1+\rho_3
\end{array} \right)
\end{displaymath}
```

图 5.14　矩阵分隔命令行

$$X = \left(\begin{array}{c|c|c} \rho_1 & \rho_2 & \frac{\rho_3}{3} \\ \hline \rho_1 & 0 & \frac{\rho_3}{2} \\ \hline \rho_1 & \rho_2 & \frac{\rho_3}{2} \\ \hline 0 & 0 & -\rho_1 + \rho_3 \end{array}\right)$$

图 5.15　矩阵分隔效果

对于多行的公式也存在编号问题,情况相对来说比较复杂。对于分布于几行的公式或者方程组(equation system),可以使用 eqnarray 和 eqnarray * 环境来代替 equation。在 eqnarray 中,每一行都会有一个方程编号,在 eqnarray * 中不对方程进行编号。另外

使用\nonumber 命令也可以阻止 LaTeX 为公式生成一个编号。

eqnarray 和 eqnarray* 环境类似于 {rcl} 形式的三列表格。中间的一列可以用作等号或不等号，或者其他看起来适合的符号，使用 \\ 命令分行。图 5.16 给出了使用 eqnarray 多行公式编号命令行，图 5.17 是之后的运行效果。

```
\begin{eqnarray}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2} \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
NU^{--}=-\rho_1+\rho_3
\end{eqnarray}
```

图 5.16 使用 eqnarray 多行公式编号命令行

$$NU^{+} = \rho_1 + \rho_2 + \frac{\rho_3}{3} \quad (1)$$

$$NU = \rho_1 + \frac{\rho_3}{2} \quad (2)$$

$$NU^{-} = -\rho_1 + \rho_2 + \frac{\rho_3}{2} \quad (3)$$

$$NU^{--} = -\rho_1 + \rho_3 \quad (4)$$

图 5.17 使用 eqnarray 多行公式编号效果

当多行公式都是独立公式时，这种编号方式比较合适，但是当多行公式同属一个公式，只是分行输入时，给每一行一个编号，显然不合适。因此需要为多行公式定义一个编号。

可以使用\nonumber，在前面几行的后面加上\nonumber，如图 5.18 所示，之后的效果如图 5.19 所示。

```
\begin{eqnarray}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3}\nonumber \\
NU=\rho_1+\frac{\rho_3}{2} \nonumber \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\nonumber \\
NU^{--}=-\rho_1+\rho_3
\end{eqnarray}
```

图 5.18 使用 nonumber 为多行定义一个编号

$$
\begin{aligned}
NU^{+} &= \rho_1 + \rho_2 + \frac{\rho_3}{3} \\
NU &= \rho_1 + \frac{\rho_3}{2} \\
NU^{-} &= -\rho_1 + \rho_2 + \frac{\rho_3}{2} \\
NU^{--} &= -\rho_1 + \rho_3
\end{aligned} \tag{1}
$$

图 5.19　使用 nonumber 多行公式编号效果

但是图 5.19 的效果不好，用户希望将编号显示在公式的中间，可以使用 array 实现。命令行如图 5.20 所示，效果如图 5.21 所示。

```
\begin{eqnarray}
\begin{array}{l}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2}\\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2} \\
NU^{--}=-\rho_1+\rho_3
\end{array}
\end{eqnarray}
```

图 5.20　使用 array 为多方定义一个编号

$$
\begin{array}{l}
NU^{+} = \rho_1 + \rho_2 + \frac{\rho_3}{3} \\
NU = \rho_1 + \frac{\rho_3}{2} \\
NU^{-} = -\rho_1 + \rho_2 + \frac{\rho_3}{2} \\
NU^{--} = -\rho_1 + \rho_3
\end{array} \tag{1}
$$

图 5.21　使用 array 多行公式编号效果

在输入数学公式时，需要注意文字在数学环境和正文环境中的不同之处。例如在数学环境中：

(1) 空格和分行都将被忽略，也就是说，在数学环境下，LaTeX 不区分是否有空格，即使输入空格，LaTeX 也不识别。如果确实需要在数学环境中输入空格，则可以由特殊的命令来得到，例如\、\quad 或 \qquad。

(2) 不允许有空行，每个公式中只能有一个段落。

(3) 每个字符都将被看作是一个变量名并以此来排版。如果希望在公式中出现普通的文本(使用正体字并可以有空格),那么用户必须使用命令\textrm{…} 来输入这些文本。命令行如图 5.22 所示,效果如图 5.23 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}) \textrm{ for all } b\in \{0,1\}.
\end{equation}}
```

图 5.22 公式中文本正体显示命令行

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}) \text{ for all } b \in \{0,1\}. \quad (1)$$

图 5.23 公式中文本正体显示效果图

一般的字符,在 amsmath 宏包中都可以找到,但是有些字体需要用到其他宏包。例如,表示实数集合的“空心粗体”(blackboard bold),这种字体可用 amsfonts 或 amssymb 宏包中的命令\mathbb 来得到。例如,将图 5.23 中的{0,1}变为空心 R,就可以使用图 5.24 所示的命令行,最后的效果如图 5.25 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}) \textrm{ for all } b\in \mathbb{R}.
\end{equation}}
```

图 5.24 空心 R 的命令行

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}) \text{ for all } b \in \mathbb{R}. \quad (1)$$

图 5.25 空心 R 的效果

5.3　一些基本的数学符号

(1) 小写希腊字母(Lowercase Greek letters)的输入命令为 \alpha、\beta、\gamma…，大写希腊字母只要把命令的首字母大写即可，输入命令为 \Gamma、\Delta…

例如，希腊字母 λ、ξ、π、μ、Φ、Ω 所对应的命令如下：$ \lambda、\xi、\pi、\mu、\Phi、\Omega $ 。

(2) 平方根(square root)的输入命令为 \sqrt，n 次方根相应地为 \sqrt[n]。方根符号的大小由 LaTeX 自动加以调整，也可用 \surd 仅给出符号。

(3) 水平线命令 \overline 和 \underline 在表达式的上、下方画出水平线。

(4) 大括号命令 \overbrace 和 \underbrace 在表达式的上、下方给出一水平的大括号。

(5) 数学重音符号可覆盖多个字符的宽，重音符号可由 \widetilde 和 \widehat 等得到。

(6) 向量 (Vectors)通常用上方有小箭头(arrow symbols)的变量表示。这可由 \vec 得到。另两个命令 \overrightarrow 和 \overleftarrow 在定义从 A 到 B 的向量时非常有用。

一般情况下，乘法算式中的圆点符可以省略。然而有时为了帮助解读复杂的公式，也有必要用命令 \cdot 将圆点符表示出来。

(7) 函数名通常用罗马字体正体排版，而不是像变量名一样用意大利体排版。因此，LaTeX 提供下述命令来排版最重要的一些函数名。

```
\arccos \cos \csc \exp \ker \limsup \min
\arcsin \cosh \deg \gcd \lg \ln \Pr
\arctan \cot \det \hom \lim \log \sec
\arg \coth \dim \inf \liminf \max \sin
\sinh \sup \tan \tanh
```

(8) 排版模函数(modulo function)有两个命令：\bmod 用于二元运算符 $a \bmod b$，

\pmod 用于表达式，例如 $x \equiv a \pmod{b}$。

(9) 分数(fraction)使用 \frac{…}{…} 排版。一般来说，1/2 这种形式更受欢迎，因为对于少量的分式，它看起来更好些。

(10) 积分运算符 $\int$ (integral operator)用 \int 来生成。

(11) 求和运算符 $\sum$ (sum operator)由 \sum 生成。

(12) 乘积运算符×(product operator)由 \prod 生成。上限和下限用“ ^”和“_”来生成，类似于上标和下标。

(13) 对于括号(braces)和其他分隔符(delimiters)，在 TEX 中有各种各样的符号(例如 $[h\ k\ l]$)。圆括号和方括号可以用相应的键输入。花括号用 \{。其他的分隔符用专门命令(如 \updownarrow)来生成。

(14) 如果将命令 \left 放在分隔符前，TeX 会自动决定分隔符的正确大小。注意必须用对应的右分隔符 \right 来关闭每一个左分隔符 \left，并且只有当这两个分隔符排在同一行时大小才会被正确确定。

(15) 将 3 个圆点(three dots)输入公式可以使用几种命令。\ldots 将点排在基线上。\cdots 将它们设置为居中。除此之外，可用 \vdots 命令使其垂直，而用 \ddots 将得到对角型(diagonal dots)。

(16) 双引号。论文写作过程中可能会需要用到引号或者双引号，如果像 Word 一样直接敲入 " ，会出现错误。例如，如果在源文件中输入"emulates"，那么生成的 PDF 中会显示“emulates”。

在 LaTeX 中有专门的左引号和右引号。如果想在 PDF 中出现“emulates”的效果，应该在源文件中输入`emulates"。

需要注意的是：`不是两个单引号，而是键盘中 Tab 键上方的符号。

(17) 单引号在 LaTeX 中也不能直接用’emulates’，而是应该用`emulates’，对应的 PDF 中显示‘emulates’。注意 ` 是 Tab 键上方的符号，而’是键盘中的单引号键。

(18) 破折号和连字号。这些短划线的名称是：‘-’(连字号)、‘ - ’(短破折号)、‘——’(长破折号)和‘—’(减号)。