

9장

수식 쓰는 방법

TeX의 최대장점 중의 하나는 수식이 아름답게 표현된다는 것이다. TeX이 원래 AMS¹에서 만든 것이기 때문에 각종 잡지나 책에 나오는 수학기호를 기본적으로 제공해준다.

9.1 간단한 수식 표현

수식 표현은 주로 아래위 첨자와 분수의 표현, 근호(root), 생략 기호와 수학 기호, log 등으로 이루어져 있다. 이 수식 모드에서의 영문자들은 이탤릭체로 나타나고 일반 텍스트보다 간격이 좀 넓다. 또한 영문자 사이의 빈 공간은 무시된다. 한가지 알아 둘 것은 수식에서는 한글을 쓸 수 없다는 것이다. 한글을 쓸 때에는 `\mbox` 명령을 사용한 텍스트로 입력한다.

1. math environment : `$ equation-text $` 형태

문장 중간에 수식이 나타날 때 사용하는 방법이다. $\sum_{n=0}^{\infty} x_n$ 과 같은 수식이 나타날 때는 수식의 앞뒤에 `$` 를 붙여주면 된다. 간단한 경우에는 `$ 수식 $` 로도 되지만 나중에 애러가 나오기 쉬운 부분이므로 주의하기

¹Americal Mathematical Society: 미국수학회

바란다. 이와 같은 기능으로 `\(equation-text \)` 또는 `\begin{math} equation-text \end{math}`라고 해도 된다.

2. `displaymath` environment: `\[equation-text \]` 아래와 같이 수식을 나타낼 때 사용하는 방법이다. 별 옵션이 없으면 가운데에 수식이 나타난다.

$$\sum_{n=0}^{\infty} x_n$$

이는 앞뒤에 `$$` 를 붙여주거나 `\[equation-text \]` 또는 `\begin{displaymath} equation-text \end{displaymath}` 와 같이 할 수 있다.

3. `equation` environment: `\begin{equation} equation-text \end{equation}` 이는 위에서 설명한 `displaymath`와 같다. 하지만 여기에서는 수식의 오른쪽 끝에 수식번호를 붙여준다. 아래의 수식과 같이 나오게 된다.

$$\sum_{n=0}^{\infty} x_n \tag{9.1}$$

이 번호는 각 장(chapter)이나 article에서 자동으로 증가하게 된다.

참고로 `fleqn` 옵션은 `{displaymath}`나 `{equation}` environment에서 수식을 왼쪽으로 붙여준다. 이를 위해서는 `\documentstyle` 명령을 다음과 같이 써주면 된다.

```
\documentstyle[12pt,fleqn]{article}
```

아래의 예는 자주 쓰이는 간단한 수식들의 이용 실례이다.

- 상하첨자

<i>output</i>	<i>input</i>
x_2	<code>\$x_2\$</code>
$2y_3$	<code>\$2y_3\$</code>
x_2y_3	<code>\$x_2y_3\$</code>
$x = y_{4z}$	<code>\$x=y_{4z}\$</code>
x^2	<code>\$x^2\$</code>
x^2y^3	<code>\$x^2y^3\$</code>
$x = y^{4z}$	<code>\$x=y^{4z}\$</code>
$\Gamma_{\alpha\beta}^{(1)}(R)$	<code>\$\$\Gamma^{\{1\}}_{\alpha\beta}(R)\$</code>
x^{y^z}	<code>x^{y^z}</code>

- /나 `\frac`에 의한 분수의 표현

$3/4$	<code>\$3/4\$</code>
$y = \frac{1}{x}$	<code>\$y=\frac{1}{x}\$</code>

`\frac{}{}` 은 `displaymath`나 `equation` 또는 `eqnarray`에서 제대로 나타난다.

$y = \frac{x^2 + z/2}{x^3 + x + 1}$	<code>\[y=\frac{x^2+z/2}{x^3+x+1}\]</code>
$V(r) = \frac{e^{-r}}{r} - \frac{1}{r^2}$	<code>\[V(r)=\frac{e^{-r}}{r}-\frac{1}{r^2}\]</code>
$P(E_i) = e^{-\frac{E_i}{kT}}$	<code>\[P(E_i)=e^{-\frac{E_i}{kT}}\]</code>

- 근호의 표시(`\sqrt`)

$y = \sqrt{x}$	<code>\$y=\sqrt{x}\$</code>
$y = \sqrt[n]{n+1}$	<code>\$y=\sqrt[n]{n+1}\$</code>
$a_{1+1} = \sqrt{a_i^2} + \sqrt{a_i}$	<code>\$a_{1+1}=\sqrt{a_i^2}+\sqrt{a_i}\$</code>

- 생략기호: 이 중에서 `\ldots`는 수식 밖에서도 사용할 수 있다.

종류	output	input
아래	x_1, x_2, \dots, x_n	<code>\$x_1, x_2, \ldots, x_n\$</code>
가운데	$1 + 2 + \dots + 99$	<code>\$1+2+\cdots+99\$</code>
수직	:	<code>\$_\vdots\$</code>
대각선	\ddots	<code>\$_\ddots\$</code>

9.2 복잡한 수식 표현

복잡한 수식을 표현할 때 보통 ‘=’이 정렬된 형태로 나타나는 경우가 많다. \TeX 에서는 이를 자동으로 정렬해서 보기 좋게 표현하는 `environment`를 가지고 있다. 앞에서처럼 `\begin{eqnarray} equation-text \end{eqnarray}`를 써주면 된다. 아래의 예는 전형적인 실례이다.

$$\begin{aligned}
 E(X^{-n}, X > t) &\approx \frac{\beta^n}{(\alpha - 1) \dots (\alpha - n)} \int_t^\infty f(x \mid \hat{\mu}, \hat{\sigma}) dx \\
 &= \frac{\beta^n}{(\alpha - 1) \dots (\alpha - n)} \cdot \left\{ \bar{\Phi} \left(\frac{t - \hat{\mu}}{\hat{\sigma}} \right) \right\} \\
 &= \frac{\mu^n}{\prod_{i=1}^n (\mu^2 - i\sigma^2)} \cdot \left[\bar{\Phi} \left\{ \frac{\mu t - \mu^2 + n\sigma^2}{\sigma \sqrt{\mu^2 - n\sigma^2}} \right\} \right]
 \end{aligned}$$

```

\begin{eqnarray*}
E(X^{-n}, X > t) &\approx & \frac{\beta^n}{(\alpha - 1) \dots (\alpha - n)} \int_t^\infty f(x \mid \hat{\mu}, \hat{\sigma}) dx \\
&= & \frac{\beta^n}{(\alpha - 1) \dots (\alpha - n)} \cdot \left\{ \bar{\Phi} \left( \frac{t - \hat{\mu}}{\hat{\sigma}} \right) \right\} \\
&= & \frac{\mu^n}{\prod_{i=1}^n (\mu^2 - i\sigma^2)} \cdot \left[ \bar{\Phi} \left\{ \frac{\mu t - \mu^2 + n\sigma^2}{\sigma \sqrt{\mu^2 - n\sigma^2}} \right\} \right]
\end{eqnarray*}

```

때로는 우변이나 좌변이 너무 길어서 이를 나누어 주어야 하는 경우가 있다. 이 경우에는 다음과 같이 한다.

```
\begin{eqnarray*}
x & = & a - b - c - d \\
& & \mbox{} -e -f -g
\end{eqnarray*}
```

$$x = a - b - c - d - e - f - g$$

여기에서 `\mbox{}` 명령은 ‘e’ 앞에서 ‘-’부호를 쓰기위해서 빈 box를 하나 만들어주기 위한 것이다. 이를 만들어 주지않으면 ‘-’와 ‘e’가 너무 붙어버리게된다. 문장 중에서 minus 가 아닌 “negative e” 를 만들기 위해서는 `{-e}`로 써주어야 한다. `\begin{eqnarray*}`에 ‘*’를 붙인 것은 수식번호를 붙이지 말라는 뜻이다. ‘*’를 붙이지 않으면 앞에서 나왔던 equation mode와 마찬가지로 자동으로 증가하는 번호를 오른쪽에 붙여준다. 한편 긴 수식을 두줄로 잘라주는 또다른 방법이 있다.

```
\begin{eqnarray*}
\lefteqn{x+y+z+a+b = } \\
& & c + d + e + f + g
\end{eqnarray*}
```

$$x + y + z + a + b = c + d + e + f + g$$

앞서의 예와 같이 두번째 줄의 `&` 두개는 `c + d + e + f + g`를 오른쪽으로 보내주는 역할을 한다.

9.3 배열 (array)

배열을 만들기 위해서는 `array environment`를 이용해야 한다. 각 행(column)은 (l,c,r)에 의해서 왼쪽, 가운데, 오른쪽에 위치하게 된다. 각 열(row)은 `\\`에 의해서 분리된다. 열(row) 안에서 행(column)을 분리하기 위해서는 `&` 를 각 항 사이에 넣어주어야 한다. 이 `array`는 수학기호에서만 사용할 수 있다. 이는 뒤에 `tabular environment`와 유사한데 10.2를 참조하기 바란다. 아래의 예를

보자.

```

\[\begin{array}{ccc}
a & 14 & c \\
d-3 & e & f \\
g & h & \lambda \\
\end{array}\]

```

여기에서 ccc는 세개의 행이 모두 가운데 위치하게 됨을 의미한다. row 사이에 \\이 들어있고 각 행 사이에 &이 들어있음을 알 수 있다. 아래는 좀더 복잡한 배열의 예이다.

```

\label{arraymatrix}
\[\det\left|
\begin{array}{lllll}
x_0 & x_1 & x_2 & \cdots & x_n \\
x_1 & x_2 & x_3 & \cdots & x_{n+1} \\
x_2 & x_3 & x_4 & \cdots & x_{n+2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_n & x_{n+1} & x_{n+2} & \cdots & x_{2n}
\end{array}
\right| > 0\]

```

위의 예에서는 배열밖에 또다른 항이 있으며 det를 나타내기위하여 |를 배열 양쪽에 붙여 놓았다. 위의 두개의 input 예에서 보듯이 모두 displaymath 모드안에 \begin{array} \end{array}가 들어있다. 물론 이 array environment 를 equation environment에서도 실행가능하다.

여기에서 또한 짚고 넘어가야 할 것이 delimiter의 설정이다. delimiter란 양쪽에 붙는 괄호를 통칭하는 것이다. 이 delimiter를 쓸 때는 반드시 \left와 \right를 양쪽에 붙여주어야 한다. 양쪽에 붙는 것은 서로 다를 수 있으며 부

록 A.8에 나오는 모든 기호를 쓸 수 있다. 한쪽만 쓰고자 하는 경우는 반대편에 `\left`. 또는 `\right`.와 같이 점(period)을 찍어 주면 된다. 이렇게만 해주면 `delimiter`는 자동으로 높이를 조절해 준다. 아래의 예를 보고 참고하기 바란다.

```

\left[ \left( \begin{array}{c}
\left( \begin{array}{cc}
a & b \\
c & d-1 \\
e & f
\end{array} \right) \\
y \\
2z
\end{array} \right) \right]

```

$$\left[\left(\begin{array}{c} \left(\begin{array}{cc} a & b \\ c & d-1 \\ e & f \end{array} \right) \\ y \\ 2z \end{array} \right) \right]$$

또하나의 유용한 명령은 `multicolumn`, 문자 정렬, 테두리 굵기이다. 아래의 예를 보자.

```

\begin{array}{|r@{.}lr@{.}l|}
\hline
12&345 & 321&645 \\
3985&34 & 85763&346 \\
\multicolumn{2}{|c}{x} & 234&356 \\
987&82745 & \\
\multicolumn{2}{|c|}{\mbox{unk.}} \\
234&427 & \multicolumn{2}{|c|}{y} \\
\hline
\end{array}

```

12.345	321.645
3985.34	85763.346
x	234.356
987.82745	unk.
234.427	y

이 `array`는 소수점을 중심으로 정렬된 배열을 보여준다. \TeX 에서는 이를 제공하지 않기때문에 이런 차선책을 쓰는 것이다. 위의 예에서 `@{.}`으로 표시된 점을 중심으로 숫자들이 정렬된다. 물론 이자리에는 다른 글자가 들어갈 수도 있다. `\begin{array}`뒤에 붙어나오는 `|`는 세로줄을 긋는 것을 말한다. 마찬가지로 `\hline`은 가로줄을 긋는다. 즉 `\`뒤에 `\hline`을 쓰게 되면 그 아래에 가로줄을 긋는다. 어떤 내용이 하나이상의 `column`에 걸쳐있을 때 `\multicolumn{}{}{}`을 이용한다. 첫번째 `{}`에 들어가는 것이 몇개의 `col-`

umn에 걸쳐있는가를 나타낸다. 두번째 {}에 있는 것이 그 내용이 놓이는 위치이고(r,c,l), 세번째가 들어갈 내용이다. \mbox{} 명령은 {} 안의 내용이 일반적인 text mode (roman type)으로 처리된다는 것을 의미한다.

9.4 기호및 기타

TeX에서 쓸 수 있는 수학기호들은 수식모드에서만 쓸 수 있다. 예를 들면 α 를 쓰기 위해서는 수식모드임을 나타내는 \$를 앞뒤에 붙여 \$\alpha\$라고 써야한다. 또한 반드시 \다음에 나타나야 한다. 예를 들어 $y^2 \neq 4$ 를 나타내기 위해서는 $y^2 \not= 4$ 라고 써야 한다. 한편 TeX는 네개의 간격 조정 명령을 제공한다. \, (조금), \: (중간정도), \; (많이), \! (약간좁힌다)이다.

예)

input \$A\,B\C\D\!E\$

output A B C D E

책 뒤의 부록에 ‘log 류’ 함수 list가 나와 있다. 이들은 수식중에서 자동으로 roman으로 나타난다. 수식중에서 다른 문자들을 roman으로 나타내기 위해서는 간단히 \rm을 써주면 된다. 예를 들면 {\rm sinc}와 같이 쓰면 된다.

한편 이 기호들을 쌓는 방법이 있다. stackrel{}{}이 그것인데 예를 들자면 $\overrightarrow{+}$ 와 같은 symbol은 $\stackrel{\rightarrow}{+}$ 이라고 입력함으로 써 가능하다. 수식의 위아래에 줄을 긋기 위해서는 \overline{ }이나 \underline{ }을 쓰면된다. 이는 수식뿐 아니라 일반 text mode에서도 가능하다. 다음과 같은 \widehat{ }과 \widetilde{ }과 brace기능도 있다.

$\widehat{A+B}$ =
 \widetilde{AB} \\
 $\overbrace{y^2+}$
 $\underbrace{y+z+3}$

$$\widehat{A+B} = \widetilde{AB}$$

$$\overbrace{y^2+y+z+3}$$

\choose도 알아두면 유용한 명령이다. 이는 다음과 같이 쓴다.

$$\$_m C_n = \{m \text{ \choose n}\} \quad {}_m C_n = \binom{m}{n}$$

TeX에서는 calligraphic문자를 수식에서 제공하는데 alphabet 대문자 26자가 있다. 이를 쓰려면 `\cal F`와 같이 하면 된다.

$$\left[\{\cal F\} = \frac{\partial \mathcal{V}}{\partial r} \right]$$

9.5 수식의 사용의 실례

이들을 위한 input은 그림과 같다.

9.6 수식출력의 예

이번 보기에서는 theorem, corollary, lemma 등의 사용 예를 보여준다.

수식을 출력할 수 있는 환경에는 세 가지가 있는데 `\(나 $로 시작 하여 \)나 $로 끝나는 수식환경과 \[나 $$로 시작하여 \]나 $$로 끝나는 수식 전시환경과 이 밖에 등식 환경이 있다. 수식환경은 문장속에서 수식을 쓸 때 사용되며 수식 전시환경은 주로 긴 수식이나 강조하고 싶은 수식을 새로운 줄의 중앙에 전시할 필요가 있을 때 사용된다. 등식환경은 여러 줄의 등식의줄을 맞추어 쓸 필요가 있을 때 주로 사용되며 식의 오른쪽에 자동적으로 참조 번호를 붙여준다. 이러한 수식환경 안에서는 모든 알파벳이 이탤릭체로 되는 것이 기본이며 띄어쓰기가 무시되므로 보통의 문장을 이러한 환경안에서 쓰기 위해서는 \mbox라는 명령의 인수으로써 그 문장을 입력해야 한다.`

9.6.1 기초적인 수식

첨수를 표시하기 위한 올려쓰기와 내려쓰기는 각각 `^`와 `_`를 사용하여 보기 9.3의 줄 19에서처럼 쓰면 되는데 여기서 `\int`는 적분기호를 나타내고 아래쓰

보기 9.1: 수식의 사용의 실례

1.

$$w^* = \lim_{n \rightarrow \infty} \frac{f^n(\Theta_o) - \Theta_o}{n}$$

2.

$$\ddot{x} + k\dot{x} + \nabla V(x) = A \cos \omega t$$

3.

$$Q = \eta e^{i\phi(x)}, Q^* = \eta e^{-i\phi(x)}$$

4.

$$R_n \leq \frac{x^n}{n!} \times m(m-1) \cdots (m-n+1)$$

5.

$$a_n \oint (z - z_0)^n dz = a_n \frac{(z - z_0)^{n+1}}{n+1} \Big|_{z_1}^{z_1}$$

6.

$$\sinh z = \sum_{n=1, \text{odd}}^{\infty} \frac{z^n}{n!} = \sum_{s=0}^{\infty} \frac{z^{2s+1}}{(2s+1)!}$$

7.

$$\underbrace{a + b + \cdots + y + z}_{26} \quad A \xrightarrow{a'} B \xrightarrow{b'} C$$

8.

$$\begin{aligned} \mathbf{U}(\alpha, \beta, \gamma) &= \begin{pmatrix} e^{i\gamma/2} & 0 \\ 0 & e^{-i\gamma/2} \end{pmatrix} \begin{pmatrix} \cos \beta/2 & \sin \beta/2 \\ -\sin \beta/2 & \cos \beta/2 \end{pmatrix} \begin{pmatrix} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2} \end{pmatrix} \\ &= \begin{pmatrix} e^{i(\alpha+\gamma)/2} \cos \beta/2 & e^{i(\alpha-\gamma)/2} \sin \beta/2 \\ -e^{-i(\alpha-\gamma)/2} \sin \beta/2 & e^{-i(\alpha+\gamma)/2} \cos \beta/2 \end{pmatrix} \end{aligned}$$

9.

$T(23)$		E	$3C_2$	$4C_3$	$4C'_3$
$\left. \begin{matrix} (R_x, R_y, R_z) \\ (x, y, z) \end{matrix} \right\}$	A	1	1	1	1
	E	$\begin{cases} 1 & 1 & \omega & \omega^2 \\ 1 & 1 & \omega^2 & \omega \end{cases}$			
	T	3	-1	0	0

$$\omega = e^{2\pi i/3}$$

보기 9.2: 입력화일

```

1  1. \[ w^* = \lim_{n \rightarrow \infty} \frac{f^n(\Theta_0) - \Theta_0}{n} \]
2  2. \[ \ddot{x} + k\dot{x} + \nabla V(x) = A \cos(\omega t) \]
3  3. \[ Q = \eta e^{i\phi(x)}, Q^* = \eta e^{-i\phi(x)} \]
4  4. \[ R_n \leq \frac{x^n}{n!} \times m(m-1) \cdots (m-n+1) \]
5  5. \[ a_n \int (z-z_0)^n dz = \left. a_n \frac{(z-z_0)^{n+1}}{n+1} \right|_{z_1}^{z_2} \]
6  6. \[ \sinh z = \sum_{n=1, \text{odd}}^{\infty} \frac{z^n}{n!} = \sum_{s=0}^{\infty} \frac{z^{2s+1}}{(2s+1)!} \]
7  7. \[ \underbrace{a + \overbrace{b \cdots y}^{24}} + z_{26} \]
8  A \stackrel{a}{\rightarrow} B \stackrel{b}{\rightarrow} C \]
9  8. \begin{eqnarray*}
10  \{ \sf U(\alpha, \beta, \gamma) \&\& \left( \begin{array}{cc}
11  e^{i\gamma/2} & 0 \\
12  0 & e^{-i\gamma/2} \end{array} \right) \\
13  \end{array} \right) \\
14  \left( \begin{array}{cc}
15  \cos \beta/2 & \sin \beta/2 \\
16  -\sin \beta/2 & \cos \beta/2 \end{array} \right) \\
17  \end{array} \right) \\
18  \left( \begin{array}{cc}
19  e^{i\alpha/2} & 0 \\
20  0 & e^{-i\alpha/2} \end{array} \right) \\
21  \end{array} \right) \\
22  \&\& \left( \begin{array}{cc}
23  e^{i(\alpha+\gamma)/2} \cos \beta/2 & e^{i(\alpha-\gamma)/2} \sin \beta/2 \\
24  -e^{-i(\alpha-\gamma)/2} \sin \beta/2 & e^{-i(\alpha+\gamma)/2} \cos \beta/2 \end{array} \right) \\
25  \end{array} \right) \\
26  \end{eqnarray*}
27  9. \begin{array}{c|c|ccc}
28  \hline
29  \multicolumn{2}{c}{T(23)} & E & 3C_2 & 4C_3 & 4C'_3 \\
30  \hline
31  & A & 1 & 1 & 1 & 1 \\
32  & E & \left( \begin{array}{c} 1 \\ 1 \end{array} \right) & \right. & & \\
33  & \begin{array}{c} 1 \\ 1 \end{array} & & & & \\
34  & \begin{array}{c} \omega \\ \omega^2 \end{array} & & & & \\
35  & \begin{array}{c} \omega^2 \\ \omega \end{array} & & & & \\
36  & \left. \begin{array}{l} (R_x, R_y, R_z) \\ (x, y, z) \end{array} \right) & & & & \\
37  & \end{array} \right) \&\& T & 3 & -1 & 0 & 0 \\
38  \hline
39  \end{array} \]
40  \&\& \omega = e^{2\pi i/3}
41  \end{eqnarray*}

```

```

1  % 수식이 포함된 한글 LaTeX 입력 파일의 예 --- 한국과학기술원 수학연구소
2
3  \documentstyle[11pt]{article}
4  \newtheorem{theorem}{정리}% [section]
5  \newtheorem{corollary}[theorem]{따름정리}
6  \newtheorem{lemma}[theorem]{보조정리}
7  \newenvironment{pf}{\medskip\noindent\bf 증명.} {\enspace\hfill\newline\smallskip}
8  \setcounter{section}{2}
9
10 \begin{document}
11 \section{월리스의 곱}
12 우리의 처음 목표는 월리스의 곱으로 알려진 다음 극한을 구하는 것이다.
13 \begin{theorem} $$\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 4 \cdot 6 \cdot 8 \cdots 2n}{1 \cdot 3 \cdot 5 \cdots (2n-1)}$$
14 \end{theorem}
15 \begin{pf} 이 증명은 다음과 같이 연습문제로 남겨둔다.
16 \begin{enumerate}
17 \item sine 함수의 거듭제곱의 적분에 관한 점화식을 사용하여 다음을 증명하라.
18   $$\int_0^{\pi/2} \sin^{2n} x \, dx = \frac{2n-1}{2n} \frac{2n-3}{2n-2} \cdots \frac{1}{2} \frac{\pi}{2},$$
19   \quad \int_0^{\pi/2} \sin^{2n+1} x \, dx = \frac{2n}{2n+1} \frac{2n-2}{2n-1} \cdots \frac{2}{3} \frac{\pi}{2}.$
20 \item  $a=1, 2, 3, \dots$  에 대하여  $\sin^n x$  가 감소한다는 사실과 정의의 정적분들을
21 이용하여 다음 부등식이 성립함을 보여라.
22   $$1 \leq \int_0^{\pi/2} \sin^{2n-1} x \, dx \leq \int_0^{\pi/2} \sin^{2n+1} x \, dx \leq 1 + \frac{1}{2n}$$
23 \item 구간  $[0, \pi/2]$ 에서의  $\sin^{2n} x$ 와  $\sin^{2n+1} x$ 의 정적분의 비를 취하여
24 월리스의 곱을 유도하라.
25 \end{enumerate}
26 \end{pf}
27 \vspace{-25pt}
28 \begin{corollary} $$\lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n}}{(2n)! n^{1/2}} = \sqrt{2\pi}$$
29 \end{corollary}
30 \begin{pf} 월리스의 곱은  $\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 4 \cdot 6 \cdots (2n)}{1 \cdot 3 \cdot 5 \cdots (2n-1)} \frac{1}{2n}$ 와 같이 바꾸어 쓸 수 있다.
31 이 식의 양변의 제곱근을 취하면 원하는 식이 얻어진다.
32 \end{pf}
33 마지막으로 다음 식이 성립함을 보이고 스텔링의 식에 들어있는 상수  $c$ 의 값이
34  $\frac{1}{\sqrt{2\pi}}$ 임을 보여라.
35 \begin{eqnarray*}
36 c &= \lim_{n \rightarrow \infty} \frac{(2n)^{2n+\frac{1}{2}} e^{-2n}}{(2n)!} \\
37 &= \lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n} \sqrt{2}}{(2n)! n^{1/2}} \\
38 &= \left[ \frac{n^{n+\frac{1}{2}} e^{-n}}{n!} \right]^2 = \sqrt{2\pi} c^2
39 \end{eqnarray*}
40 \end{pf}
41 \end{document}

```

보기 9.3: 수식이 포함된 L^AT_EX 화일의 예

3 윌리스의 곱

우리의 처음 목표는 윌리스의 곱으로 알려진 다음 극한을 구하는 것이다.

정리 1

$$\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdots \frac{2n}{2n-1} \frac{2n}{2n+1}}$$

증명. 이 증명은 다음과 같이 연습문제로 남겨둔다.

1. sine 함수의 거듭제곱의 적분에 관한 점화식을 사용하여 다음을 증명하라.

$$\int_0^{\pi/2} \sin^{2n} x dx = \frac{2n-1}{2n} \frac{2n-3}{2n-1} \cdots \frac{1}{2} \frac{\pi}{2}, \quad \int_0^{\pi/2} \sin^{2n+1} x dx = \frac{2n}{2n+1} \frac{2n-2}{2n-1} \cdots \frac{1}{2} \frac{1}{2}$$

2. $a = 1, 2, 3, \dots$ 에 대하여 $\sin^a x$ 가 감소한다는 사실과 정의 정적분들을 이용하여 다음 부등식이 성립함을 보여라.

$$1 \leq \frac{\int_0^{\pi/2} \sin^{2n-1} x dx}{\int_0^{\pi/2} \sin^{2n+1} x dx} \leq 1 + \frac{1}{2n}$$

3. 구간 $[0, \pi/2]$ 에서의 $\sin^{2n} x$ 와 $\sin^{2n+1} x$ 의 정적분의 비를 취하여 윌리스의 곱을 유도하라.

따름정리 2

$$\lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n}}{(2n)! n^{1/2}} = \pi^{1/2}$$

증명. 윌리스의 곱은 $\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2^2 4^2 \cdots (2n-2)^2}{3^2 5^2 \cdots (2n-1)^2} 2n$ 와 같이 바꾸어 쓸 수 있다. 이 식의 양변의 제곱근을 취하면 원하는 식이 얻어진다. 마지막으로 다음 식이 성립함을 보이고 스텔링의 식에 들어있는 상수 c 의 값이 $1/\sqrt{2\pi}$ 임을 보여라.

$$\begin{aligned} c &= \lim_{n \rightarrow \infty} \frac{(2n)^{2n+\frac{1}{2}} e^{-2n}}{(2n)!} \\ &= \lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n} \sqrt{2}}{(2n)! n^{1/2}} \left[\frac{n^{n+\frac{1}{2}} e^{-n}}{n!} \right]^2 = \sqrt{2\pi} c^2 \end{aligned}$$

보기 9.4: 수식이 포함된 파일이 처리되어 출력되어진 모습

기 0과 올려쓰기 $\pi/2$ 가 사용되고 있다. 또한 줄 32에서는 2^2 등의 입력 방법이 나타나있다.

분수는 줄 29에서와 같이 /를 사용하여 $1/2$ 처럼 입력하거나 줄 13에서처럼 `\frac` 명령을 사용하면 되는데 `\frac`는 다음에 나오는 두개의 인수를 취하여 처음 것은 분자 다음 것은 분모로 처리한다.

근호는 줄 36에서처럼 `\sqrt`를 사용한다. 다만 n 승근을 표기하기 위하여는 `\sqrt[n]` 명령을 대신 사용하면된다.

생략을 표시하기 위한 점들은 줄 14와 같이 `\cdots`를 사용하면 줄 중간에 점 세개가 찍히고 줄 21에서처럼 `\ldots`를 사용하면 줄 아래 쪽으로 점 세개가 찍힌다. 이 밖에도 세로로 점이 찍히는 `\vdots`와 대각선 방향으로 점이 찍히는 `\ddots`가 있다.

9.6.2 수학기호

수학기호에는 많은 것들이 있기 때문에 여기서 모두 다루지는 않고 한가지씩의 예를 들어봄으로써 대신하고자 한다. 여러분은 이를 출력하기 위하여 원 매뉴얼을 참고하기 바란다.

그리스문자는 문자의 영문이름을 `\`로 시작하여 쓰면 되는데 줄 13에서 `\pi`는 π 를 출력한다.

많은 연산자 기호를 위한 한가지 예를 들면 \pm 은 `\pm`으로 얻는다.

관계자 기호 전후에는 자동으로 약간의 공백이 들어간다. 관계자 기호를 위한 한 예는 줄 23의 `\le`이다. 이 `\le`는 \leq 를 얻는다.

많은 화살표를 위한 명령이 있는데 그 중 한 예는 `\rightarrow`이다. 이 명령은 \Rightarrow 를 얻는다.

함수기호를 위한 명령은 줄 13의 `\lim`과 같은 것이 있는데 이것은 수식 안이지만 명조체로 `lim`을 출력한다. 이 함수기호에 수식 전시환경에서 올려쓰기나 내려쓰기를 하면 보기 9.4에서 보듯이 그 기호의 위나 아래에 (기호옆 위나 아래가 아닌) 첨자가 붙는다.

등식환경은 `\begin{eqnarray*}` 명령과 `\end{eqnarray*}` 명령을 사용하여 얻을 수 있다. 이 때 줄을 맞추고자 하는 곳에 `&`를 써주면 된다. 이 예는 [줄 37](#)에서 [줄 41](#)을 참고하기 바란다.

부록 A

오늘날의 수식 (권장) 입력방식과의 차이

written by Nova de Hi.

이 문서는 WINLATEX 1.0의 안내문서에서 9장 <수식 쓰는 방법>을 재현한 것이다. 표기법과 띄어쓰기는 원본대로 표기하는 것을 원칙으로 했다. 이 문서가 작성된 때는 1991~1994년 사이였던 것으로 보이며, 9장의 전반부는 yahTeX <MANUEL> 과 내용이 완전히 동일한 것으로 볼 때 당시까지 통용되던 매뉴얼을 이것저것 합친 것으로 보인다. 그 결과, 이 문서의 작성자는 적어도 두 명으로서 내용이 중복되는 이유는 그것에 기인하였다. 이 문서에는 ‘file’, ‘화일’, ‘파일’ 세 가지 표기가 혼재하고 있다. 그것은 문서 작성자가 적어도 둘 이상이라는 심증을 확인해주는 증거이다.

이 문서는 원본의 예스런 느낌을 살리기 위해 폰트를 백묵으로 하고 되도록 당시 문서의 외양에 가깝게 꾸몄다. 클래스는 report. 그러나 부득이하거나 그렇게 하는 것이 좋겠다고 판단한 부분(예를 들면 verbatim 텍스트에 대하여 label을 붙이고 이를 \ref 하는 것 등)에서는 modern한 방법을 채용하여 코딩하였다.

부록에서는 이 문서에서 설명하는 내용 중에서 현재의 L^AT_EX 환경에 맞지

않거나 또는 권장하지 않게 변한 사항을 지적하려 한다.

A.1 eqnarray의 폐기

가장 큰 변화는 eqnarray 환경에 대하여 “폐기” 판정을 내린 것이다. 이 판정이 권위를 가지고 이루어지는 것도 아니며 L^AT_EX에서 eqnarray가 축출된 것도 아니라서 여전히 이 환경을 써서 문서를 만들어도 컴파일에 문제가 발생하지는 않는다. 그러나 이제 이 환경은 결단코 써서는 안 되며, 이미 이렇게 입력된 것이 있다면 마땅히 모두 수작업 수정해야 한다. 실로 eqnarray는 L^AT_EX의 흑역사 같은 것이다.

lshort이라는 유명한 문서는 왜 eqnarray를 쓰면 곤란한지를 무려 한 개 절을 할애하여 상세히 설명하고 있다. 여러 이유가 있지만 그 중 가장 거슬리는 것은 다음 두 가지이다.

- 등호 좌우의 간격이 너무 넓어서 보기 실다는 점
- 수식이 수식번호를 먹어들어가서 겹쳐 찍히는 것

eqnarray를 대신하는 것으로 lshort 문서는 IEEEeqnarray를 추천하고 있지만 일반적으로 볼 때 $\mathcal{A}\mathcal{M}\mathcal{S}$ -math의 align 환경으로 바꾸는 것이 더 나은 듯하다. 다만 이 경우 컬럼 정렬자(&) 두 개 중에 하나를 없애야 한다는 문제가 있기는 하다. 이밖에도 몇 가지 추가적인 문제가 있는 것이, 한쪽 변만이 길어지는 경우를 잘 처리하지 못하는 것도 거기에 속한다.

eqnarray는 이름이 잘 보여주듯이 array를 이용하여 여러 줄 수식을 조판하는 환경이다. 어떤 의미에서 기발하다고 하겠으나 현재의 관점에서 보자면 특히 초심자의 경우에 이 환경을 배워서 쓰는 것은 아무런 이득이 없다고 본다.

이왕 말이 나온 김에 수식 번호와 관련해서 align 사용법에 대해 간략히 소개한다.

수식번호가 붙지 않을 때 단순히 align* 환경을 쓰는 것으로 해결한다.

수식번호가 특정 행에만 붙을 때 이 때에는 align 환경을 쓴다. 수식번호를 생략하는 행에는 \nonumber를 적어준다.

```

\begin{align}
a &= b+c+d \nonumber \\
&= f(x) \pm g(x) \label{test:1} \\
&= l + m - q - n \label{test:2} \\
\end{align}

```

$$a = b + c + d$$

$$= f(x) \pm g(x) \tag{A.1}$$

$$= l + m - q - n \tag{A.2}$$

\eqref를 써서, (A.1)과 (A.2)의 각 행을 참조할 수 있다.

여러줄 수식 전체에 하나의 수식번호를 붙일 때 이 때는 기본적으로 equation 환경을 써야 한다. 그 대신 여러줄 수식을 조판하기 위하여 aligned나 split을 쓰면 되는데, split은 어떻게 해도 페이지나누기가 되지 않는 덩어리이기 때문에 한글 책의 조판에서는 aligned를 쓰는 쪽을 선호한다. label은 행마다 붙여도 의미가 없고 equation에 붙여야 한다.

```

\begin{equation}\label{test:3}
\begin{aligned}
a &= b+c+d \nonumber \\
&= f(x) \pm g(x) \\
&= l + m - q - n
\end{aligned}\tag{A.4}
\end{equation}

```

$$a = b + c + d$$

$$= f(x) \pm g(x) \tag{A.4}$$

$$= l + m - q - n$$

식 (A.4)는 여러 줄 수식 전체에 하나의 수식번호가 붙은 것이다. aligned라는 환경으로 만들어지는 전체를 한 개의 박스라고 생각하면 알기 쉽다.

A.2 \mathcal{AMS} -Math의 행렬 환경

이 글에서 다양한 예를 들고 있는 행렬(matrix)을 array로 구현한 예들은 어떤 의미에서 낡은 것이 되었다. 오늘날 수학식이 있는 문서를 조판하는 데 있어

```
\usepackage{amsmath,amssymb,amsthm}
```

을 로드하는 것은 하나의 상식이라서, \mathcal{AMS} -Math가 제공하는 여러 함수를 자유롭게 쓰는 것을 익히는 것이 바람직하다.

대표적인 것이 행렬 환경이다. 예컨대 90 페이지에 나오는 예를 `amsmath` 방식으로 입력하자면,

```

\[\det
\begin{vmatrix}
x_0 & x_1 & x_2 & \cdots & x_n \\
& x_1 & x_2 & \cdots & x_{n+1} \\
& & x_2 & \cdots & x_{n+2} \\
& & & \ddots & \vdots \\
& & & & x_n & x_{n+1} & x_{n+2} & \cdots & x_{2n}
\end{vmatrix} > 0
\]

```

이와 같이 `amsmath`가 제공하는 여러 `matrix` 환경을 숙지하는 것이 좋겠다.

9.3 절에서 이 문서는 `row`를 열, `column`을 행이라고 하고 있는데, 이것은 현재 우리가 아는 것과 좀 다르다.

흔히 듣는 “오와 열을 맞추다”의 “伍”는 “가로 벌린 한 줄”을 가리키는 말이다. (더 근원을 파고 들면 고대 군사 편제에 대한 얘기를 해야겠지만 그것은 지금 관심사가 아니고) 다섯 사람을 나란히 세워서 한 오로 삼고 오장을 두었다. 그리고 각각의 오를 앞뒤로 세워서 그것을 ‘행(行)’이라 하였다. 이렇게 편성된 부대를 ‘향오’라고 한다. (고대의 도로 사정을 감안하면 다섯 사람 이상이 나란히 서서 행진하기는 어려웠을 것이다.) 이 ‘오’는 “대오(隊伍)”라는 말에 남아 있다. 가끔 ‘행렬’ 대신 ‘향렬’이라고 부르기를 원하는 분들이 계신데, 현재 ‘향렬’은 친족 관계를 가리킬 때만 쓰이는 말이라서 어색하게 느끼기 쉬우나, ‘行伍’를 ‘향오’라고 읽는 독법을 고려한다면 아주 근거가 없다고 못할 것이다.

‘열(列)’은 대체로 길게 늘어선 줄을 말한다. 그래서 matrix의 column을 ‘열’로 번역한 것은 좋은 선택이었다. 실제 “4열 중대”라는 말은 군대가 4개의 column으로써 하나의 ‘오’를 짓고 앞뒤로 늘어선 상황을 가리킨다.

결론적으로 matrix $[a_{ij}]$ 에서 i 가 행(row)이고 j 가 열(column)이다. $\backslash\backslash$ 가 행구분자이고 $\&$ 가 열분리자이다. 이 문서의 설명은 뒤바뀌어 있다.

A.3 기타

91 페이지의 소수점 기준 정렬 array (또는 tabular)는 진짜 구식 방법이다. 차라리 dcolumn 패키지를 쓰는 것이 나을 것이며 그 외에도 다양한 해결책이 존재한다.

`\choose`는 amsmath의 `\binom`이 낫다. 소스를 읽을 때 덜 헛갈린다.

<code>\[m \choose n \]</code>	$\binom{m}{n}$
<code>\[\binom{m}{n} \]</code>	$\binom{m}{n}$

수식 내에서 간격을 주기 위해서 ~를 여러 개 쓰는 (좀 촌스러운) 예시가 많다. 이것은 `\quad \qqquad \hspace` 등을 이용하는 방식으로 코딩해두는 것이 나중을 위해서 좋다.

수식 내에 텍스트를 찍기 위해서 `\mbox`를 쓰라고 하고 있는데, 오늘날 보편적인 방법은 amsmath의 `\text` 명령을 쓰는 것이다. 이 문서는 NFSS가 L^AT_EX에 도입되기 전의 내용이라서 `\rm`, `\sf`, `\tt` 등의 폰트 명령이 예시되고 있으나 이것은 오늘날 deprecated 상태인 것으로 간주한다.

여담이지만 보기 9.4는 DOSBOX의 WINLATEX 가상 기계에서 컴파일하고 프린트한 결과이다.