

# L<sup>A</sup>T<sub>E</sub>X와 GNUPLOT Plotting 프로그램

David Kotz

2002년 7월 12일

역자: 쿨링팬(isyoon@pusan.ac.kr), 버전: 0.0.1

이 문서는 David Kotz의 GNUPLOT L<sup>A</sup>T<sub>E</sub>X 튜토리얼 버전 3.0에 기초하여 만들어 졌습니다. 처음에는 원문의 내용을 그대로 옮기려고 하였으나, 몇몇 예제들이 말썽을 부리고 수학적 함수구현의 예제들이 많아서, 유용하다고 생각되는 다른 몇가지 예제들을 추가하고 내용을 수정하였습니다.

## 제 1 절 소개와 역사

GNUPLOT 프로그램은 함수와 데이터 파일들을 여러가지 터미널에 표시하기 위해 Colin Kelley와 Thomas Williams에 의해 1986년 개발되었습니다. 1988년과 89년에 David는 GNUPLOT에서 L<sup>A</sup>T<sub>E</sub>X코드를 출력할 수 있는 GnuT<sub>E</sub>X이라는 새로운 L<sup>A</sup>T<sub>E</sub>X “터미널 타입”을 만들었습니다. 이것은 plot들을 L<sup>A</sup>T<sub>E</sub>X 문서에 포함될 수 있게 합니다. David는 사용자들이 출판될 수 있을 정도의 plot들을 만들 수 있도록 L<sup>A</sup>T<sub>E</sub>X 터미널에서만 지원이 되는 많은 기능들을 더하였습니다.

1989년 후반과 1990년 초에 GnuT<sub>E</sub>X과 많은 GNUPLOT들의 변형들이 GNUPLOT 2.0으로 합해졌습니다. 이것은 많은 개선점을 가지는거에 더해서 GnuT<sub>E</sub>X으로부터 나온 L<sup>A</sup>T<sub>E</sub>X 드라이버도 포함하고 있습니다. 이전의 GnuT<sub>E</sub>X의 사용자가 GNUPLOT에 어떻게 적용할 수 있는지를 알기 위해서는 제5절을 참조하십시오. GNUPLOT과 L<sup>A</sup>T<sub>E</sub>X의 사용에 관심이 있는 사용자는 다음 절부터 읽어 나가십시오.

GNUPLOT에 의해 만들어지는 L<sup>A</sup>T<sub>E</sub>X 그림은 아주 크기가 크고 복잡하기 때문에 T<sub>E</sub>X의 메모리 용량을 초과할 수 있다는 점을 독자는 인식해야만 합니다. 만약 더 큰 용량의 T<sub>E</sub>X이 사용가능하다면 그것을 사용하는 것이 현명할 것이고, 그렇지 않다면 여러분의 plot을 간단하게 만들고 문서에 필요하다면 `\clearpage`를 더하십시오.

또한 EEPIC 드라이버(eepic.c)라는 것은 L<sup>A</sup>T<sub>E</sub>X에서 EEPIC 마크로 패키지를 사용하기 위해 고안되었습니다. EEPIC은 좀 더 나은 선 그리기를 가능하게 하며 L<sup>A</sup>T<sub>E</sub>X을 좀 더 빠르게 그리고 메모리를 덜 쓰게 합니다. 더 구체적인 정보

는 제4절을 참조하십시오.

GNUPLOT과 함께 L<sup>A</sup>T<sub>E</sub>X의 plot들을 좀 더 유용하게 만드는 makefile들과 script들을 포함한 작은 패키지가 있습니다. 이것은 cs.duke.edu의 pub/gnuplot-latex.shar에서 ftp를 통해서 받을 수 있고 David이 메일로 copy를 줄 수도 있습니다(더 자세한 사항은 이 문서의 끝부분을 참조하십시오).

## 제 2 절 L<sup>A</sup>T<sub>E</sub>X를 위한 GNUPLOT 사용하기

GNUPLOT은 원래 interactive한 프로그램입니다. L<sup>A</sup>T<sub>E</sub>X를 위한 plot을 만들고자 하는 사용자들은 GNUPLOT을 interactive한 방법으로 사용하지 않을 것입니다. GNUPLOT으로부터 문서형태의 plot이 필요할 때마다, GNUPLOT이 그래픽을 표시할 수 있는 터미널에서 수행될 필요는 없습니다. 이러한 경우 그 출력물은 파일 혹은 파이프로 보내어질 수 있으며, 해당되는 출력 장치들로 보내어질 수 있습니다. 예로 들면, unixplot이라는 터미널 타입으로부터의 출력은 Unix의 plotting 표준을 해석하는 프로그램으로 보내어지게 되고, imagen과 postscript 터미널 타입은 그 언어를 이해할 수 있는 프린터로 그 출력을 보냅니다. (함께 배포되는 lasergnu라는 셸 스크립트는 GNUPLOT 명령어나 입력 파일을 받을 수 있고 그 출력을 Imagen이나 Postscript 레이저 프린터로 보낼 수 있습니다.) fig 터미널 타입은 Fig 그래픽 프로그램에서 읽혀질 수 있는 FIG 코드를 출력하는데 이는 T<sub>E</sub>X과 L<sup>A</sup>T<sub>E</sub>X들 다에서 해석되어질 수 있습니다.

우리는 이제 GNUPLOT의 상호작용적인 특성은 무시하고 GNUPLOT에게 파일로 입력을 제공할 것입니다, 예제로써,

```
gnuplot gnu.input
```

이 예제에서, GNUPLOT에게 주어질 명령들은 모두 gnu.input이라는 파일이 가지고 있습니다. 여러개의 파일명이 주어진 순서대로 GNUPLOT에게 읽혀질 수도 있습니다. 출력(하나 혹은 그 이상의 plot들)은 다른 프로그램으로 파이프되거나 파일로 리다이렉트될 수 있습니다. 그러나, 일반적으로 우리는 GNUPLOT에게 출력을 하도록 명확하게 명령을 합니다(set output 명령). GNUPLOT은 에러 메시지를 터미널(stderr)에 또한 출력합니다.

여기에 plot을 만들어 내는 첫번째 예제가 있습니다. GNUPLOT의 입력 파일은 아래에 있고, 그 출력은 그림1 입니다. 입력은 그 출력으로 L<sup>A</sup>T<sub>E</sub>X를 정의하고,  $y = \sin(x)$ 의 그래프를 주어진 파일명으로 출력합니다.

### 2.1 예제 1

```
gnu.input 내용
```

```
set terminal latex
set output "eg1.tex"
```

plot [-3.14:3.14]  $\sin(x)$

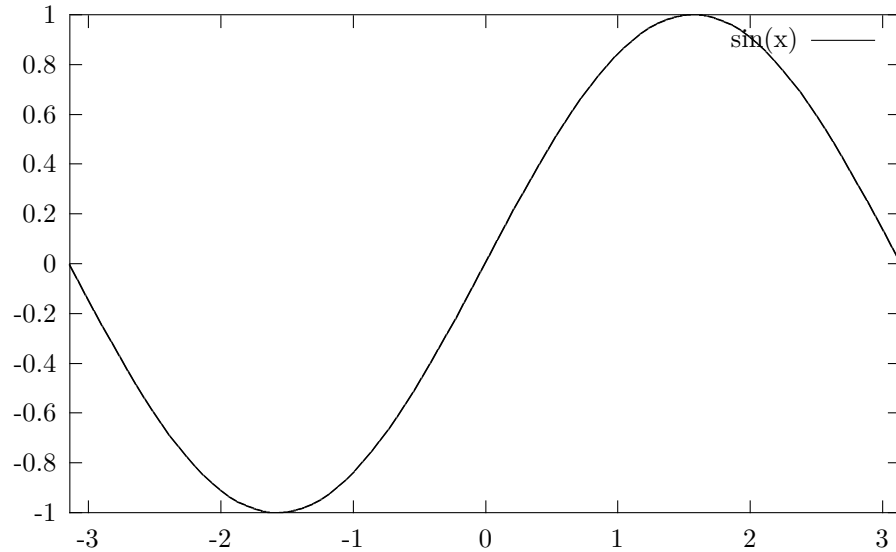


그림 1:  $\sin(x)$

gnuplot gnu.input을 수행하면 eg1.tex이라는  $\sin(x)$ 를 나타내는  $\text{T}_{\text{E}}\text{X}$ 파일이 만들어져 있습니다. 놀랍지 않습니까. 그림을 나타내기 위해 figure환경에서 `\input{eg1}`을 하면 됩니다. 그림에서 보이듯이 GNUPLOT은 좌표축과 눈금자를 그리고 y축은 자동으로 계산하며, 그리고 상단 오른쪽에 key를 나타낸거에 주의하며 보십시오(`set key`로 위치를 옮길 수도 있으며, `set nokey` 명령어로 없앨 수도 있습니다).

이것은 기본적인  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 드라이버의 형식입니다.  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 의 제한된 그리기 기능으로 인해 많은 점들로 구성된 근사적인 선을 그립니다. 이것은  $\text{T}_{\text{E}}\text{X}$ 의 구현에서 많은 메모리를 차지하게 되는데 좀 덜 메모리를 써서 점으로 표현된 선을 그릴 수 있는 다른 선 타입들도 이용 가능합니다. EEPIC 드라이버는 좀 더 적은 메모리 사용으로 채워진 선을 그립니다.

## 2.2 예제 2

이제 우리는 라벨을 가지고 plot을 꾸며볼 것입니다. 이 입력파일은 그림2를 출력합니다.

gnu.input 내용

```

set terminal latex
set output "eg2.tex"
set title "이것은  $y = \sin(x)$ 의 함수 그래프입니다."
set xlabel "이것은  $x$ 축입니다."
set ylabel "이것은\\\\\\\\ $y$ 축\\\\\\\\입니다."
plot [-3.14:3.14] sin(x)

```

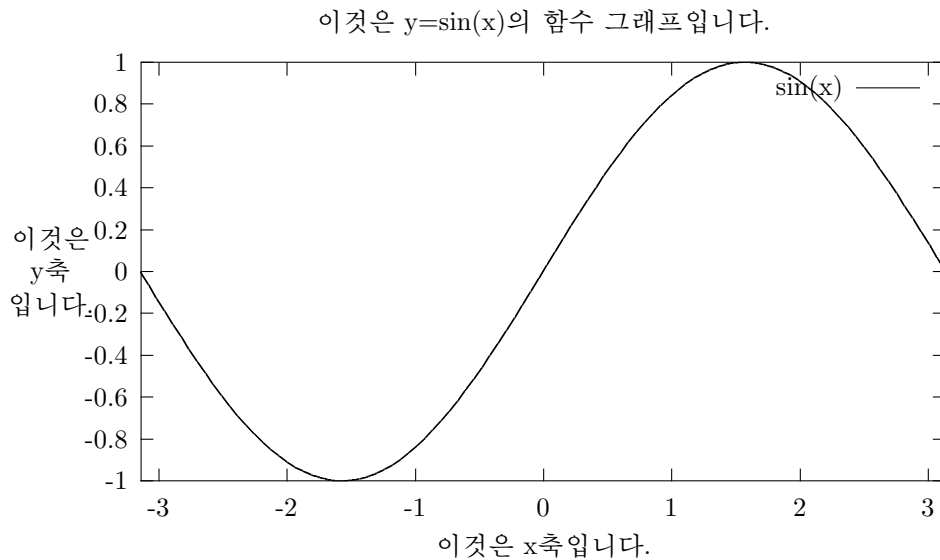


그림 2:  $\sin(x)$ 에 라벨을 첨가

입력파일의 형식은 매우 직관적이고  $y$ 축의 라벨이 길어서 행을 띄우기위해 많은 수의 `\`를 넣은 것에 주의하십시오.

### 2.3 예제 3

여러개의 그래프를 그려 보도록 하겠습니다. 실험값이 담긴 데이터 파일(mydata)을 이용하여 그려보겠습니다. 아래의 입력파일은 그림3을 출력합니다.

gnu.input 내용

```

set terminal latex
set output "eg3.tex"
set title "데이터 파일을 이용한 다중 그래프 그리기"
set xlabel " $x$  축"
set ylabel " $y$  축"
set key 7,-35

```

plot  $x$  with lines, “mydata” using 1:2 title “y1” with linespoints 1 5,  
 “mydata” using 1:3 title “y2” with linespoints 2 12

mydata 내용

```
0 0 -50
1 4 -36
2 6 -24
3 6 -14
4 4 -6
5 0 0
6 -6 4
7 -14 6
8 -24 6
9 -36 4
10 -50 0
```

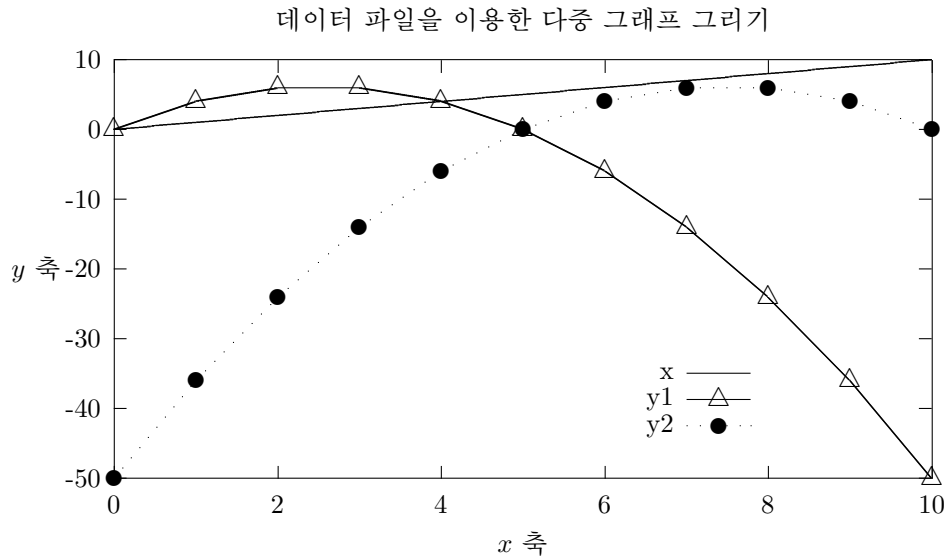


그림 3: 데이터 파일을 이용한 다중 그래프

우선  $y = x$ 의 그래프를 일반적인 선을 써서 나타냅니다. 그리고는 “mydata”라는 파일을 읽어와서 데이터의 첫번째 열과 두번째 열을 참조하여  $y1$ 의 그래프를 그립니다. 그리고 데이터의 첫번째 열과 세번째 열을 참조하여  $y2$ 의 그래프를 만들어 내는 것입니다. linespoints는 그래프를 그릴 때 어떤 선과 점을 써서 그릴것지를 나타냅니다.  $\text{\LaTeX}$ 에서는 총 6개의 선과 12개의 점을 쓸 수가 있습니다. 그래프가 많이 그려져서 기본적인 key의 자리가 적당하지 않아

$x$ 축으로 7,  $y$ 축으로 -35 만큼 옮겼습니다.

## 2.4 예제 4

이제 그래프를 상자꼴로 바꾼 출력이 그림4 입니다. 이러한 형태는 차트를 그릴때 유용할 것으로 보입니다.

gnu.input 내용

```
set terminal latex
set output "eg4.tex"
set title "박스를 이용한 그래프"
set xlabel " $x$ 축"
set ylabel " $y$ 축"
set key 4.5, 9
plot "boxdata" using 1:3:5 title "데이터1" with boxes, "boxdata"
using 2:4:5 title "데이터2" with boxes
```

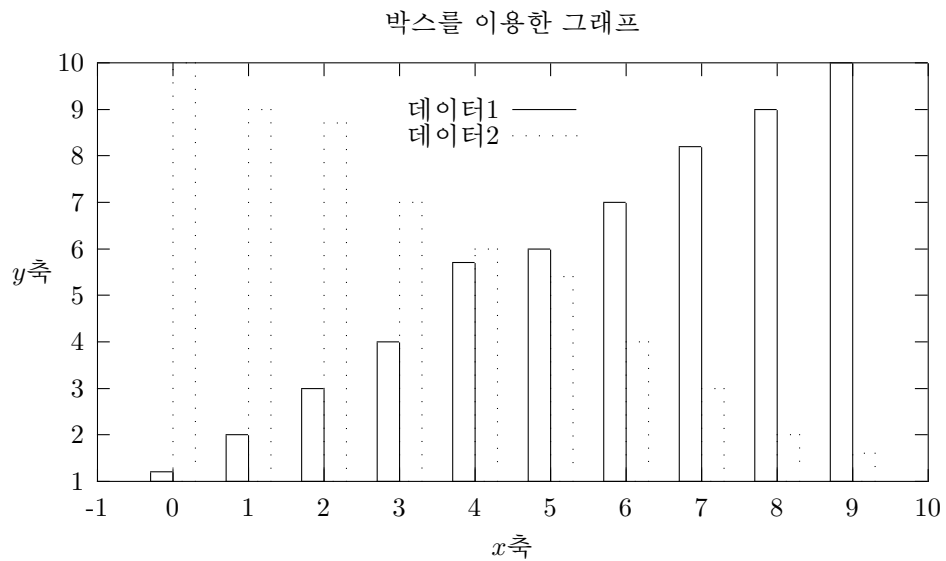


그림 4: 상자를 이용한 다중 그래프

boxdata의 내용

```
0.85 1.15 2 9 0.3
1.85 2.15 3 8.7 0.3
2.85 3.15 4 7 0.3
```

```
3.85 4.15 5.7 6 0.3
4.85 5.15 6 5.4 0.3
5.85 6.15 7 4 0.3
6.85 7.15 8.2 3 0.3
7.85 8.15 9 2 0.3
8.85 9.15 10 1.6 0.3
```

내용을 살펴보면 데이터1은 “boxdata”의 첫번째, 세번째 열의 데이터를 바탕으로 상자를 그리며 두께는 0.3입니다. 이는 “boxdata”의 다섯번째 열에 있는 값입니다. 첫번째 행을 보면 0.85가  $x$ 축 기준이니까 상자의 좌우를 합한 넓이가 0.3이 되면 정확히 0에서 상자의 우측이 접하게 됩니다. 데이터2도 이런 식으로 값을 써 나갑니다. 일단 각 상자의 축과 넓이만 잘 잡아놓으면  $y$ 축의 값만 이제 바꾸어 주면 되니까 크게 힘들지는 않을 것입니다.

### 제 3 절 요약

이번 절에서는 GNUPLOT에서 L<sup>A</sup>T<sub>E</sub>X의 기능들을 사용하는 것을 요약하겠습니다. GNUPLOT에 내리는 첫번째 명령은

```
terminal latex
```

가 되어야만 하고 출력은 파일로 나가야 합니다. 예로 들자면,

```
set output "plot.tex" 과 같이 말이죠. TEX이란 확장자만 붙으면 아무 파일명이나 상관없습니다.
```

이러한 입력파일을 가지고 GNUPLOT을 돌리게 되면 결과적으로 그래프를 나타내는 T<sub>E</sub>X파일을 가지게 되고, 이것은 다른 L<sup>A</sup>T<sub>E</sub>X문서에 포함될 수가 있습니다. 예로 들자면 아래와 같습니다.

```
\begin{figure}
\begin{center}
\input{plot}
\end{center}
\end{figure}
```

여러분들은 또한 다음과 같은 명령어들을 읽기를 원할 것입니다: set title, set xlabel, set ylabel, set key, set label, set xtics, set ytics, 그리고 set clip. 이러한 명령어들은 GNUPLOT 메뉴얼에 모두 설명되어 있습니다.

## 제 4 절 EEPIC

EEPIC은 L<sup>A</sup>T<sub>E</sub>X의 그림처리 능력을 확장한 매크로 패키지입니다. 만약 여러분들이 EPIC이나 EEPIC 매크로를 가지고 있고 dvi 해석기가 tpic \specials를 지원하면 여러분들은 L<sup>A</sup>T<sub>E</sub>X의 메모리를 절약할 수 있습니다. EEPIC환경하에서 plot.tex파일은 더 작게 될 것이며, L<sup>A</sup>T<sub>E</sub>X은 더 빨리 돌거이고, dvi 파일 또한 작아질 것입니다. 출력물의 질은 같습니다. 만약 여러분이 소스를 바꾼다면 더 많은 흥미로운 선 스타일들을 출력하게 될 것입니다.

EEPIC을 사용하기 위해서는 GNUPLOT의 터미널 타입을 latex대신에 eepic으로 바꾸어 주면 선의 스타일이 바뀌게 될 것입니다. 이를 여러분의 L<sup>A</sup>T<sub>E</sub>X문서에 넣기 위해서는 문서 스타일 옵션에 [epic,eepic]을 넣어 주십시오.

## 제 5 절 이전의 GnuT<sub>E</sub>X사용자

이전의 GnuT<sub>E</sub>X사용자들은 많은 새로운 기능들에 아마 기뻐할 것입니다 만(많은 것들이 여러분의 제안에 의한 것이었습니다!), 많은 변화가 또한 있다는걸 알게 될 것입니다. GNUPLOT이 모든 GnuT<sub>E</sub>X의 입력파일을 돌릴 수는 없을 것입니다. 몇몇의 GnuT<sub>E</sub>X의 기능들은 L<sup>A</sup>T<sub>E</sub>X드라이버에만 너무 특별하게 맞추어져 있어 GNUPLOT에는 포함되지 않았습니니다. 저는 여러분들이 새로운 GNUPLOT을 사용하기를 권합니다. 여러분의 옛날 GnuT<sub>E</sub>X1.6의 입력파일들을 GNUPLOT 3.0으로 변환하여 주는 해석기가 있습니다. 이는 cs.duke.edu의 dist/sources/gnuplot/gnu2p.tar.Z에서 ftp로 가져가실 수 있습니다. 이 파일에는 또한 GnuT<sub>E</sub>X에서 GNUPLOT으로 이행하는데에 대한 설명과 변화사항들이 들어있습니다.

## 제 6 절 연락처

여러분들이 GNUPLOT의 L<sup>A</sup>T<sub>E</sub>X드라이버에 대한 어떤 의견도 David.Kotz@Dartmouth.edu로 저에게 메일을 주십시오. 그리고 일반적인 질문들은 GNUPLOT 메일링리스트(info-gnuplot@dartmouth.edu)로 보내십시오.