gnuplot v3.7

신동원

kaien (at) kldp.org

gnuplot v3.7
지은이 신동원

펴냄 $Date: 2002/05/09 21:05:15 $

gnuplot은 명령행 입력 방식의 그래프 작성툴이다. 간단한 명령으로 2차원, 3차원의 그래프를 플롯(plot)할 수 있다. GUI방식의 Excel, Origin 등에 익숙해진 사용자들에게는 다소 불편하게 느껴질 수도 있으나 script를 이용, 약간의 프로그래밍을 해주면 Excel의 macro, Origin의 template과는 비교할 수 없는 빠른 처리 속도를 보여준다. 일반적인 리눅스 배포판을 설치했을 경우 대부분 설치가 되어 있으며 그 크기 또한 매우 작다. 그러나 그 크기가 작다고 무시하지 말도록 하자. 우리가 생각하는 거의 모든 그래프를 그릴 수 있으며 몇백만원짜리 상용프로그램에도 절대 뒤지지 않는 강력한 기능에 입을 다물지 못할 것이다.

고친 과정
고침 0.22002-05-09고친이 kaien
새로운 활용예와 고급설정의 설명 추가
고침 0.11999-10-25고친이 kaien
최초 작성

차례

그림 목록

서문

1. Linux용 공학 Application

과거 리눅스가 국내에서는 대부분의 사람들에게 서버용 운영체제로 인식되어 개인사용 자들에겐 깊이 파고들지 못하는 한계가 있었으나 최근 Gnome, KDE와 같은 GUI 환경의 개발이 활발히 이루어지면서 윈도우즈와 대적할 만한 개인 데스크탑 운영체제로 당당 히 입지를 넓혀가고 있다. 그러나 현재 국내에서 리눅스가 개인 사용자층에 널리 보급 되어 사용되는 용도 역시 윈도우즈 사용자층과 유사한 웹서핑, 파일전송, 멀티미디어 등의 제한된 용도로 사용되고 있는 것이 현실이다.

하지만 컴퓨터라는 기기의 근본적인 개발 취지가 게임, 음악감상이나 e-mail 전송과 같은 개인적인 목적이 아닌 공학계산에서 출발한 것임을 상기해 볼때 많은 아쉬움이 남는다. 물론 PC용으로 개발되어 시판되고 있는 Mathematica, Matlab, Origin 등과 같 은 수많은 공학용 프로그램들이 있으나 수백만원을 호가하는 고가의 상용프로그램일 뿐만 아니라 윈도우즈용으로 판매되고 있다.

그러나, 조금만 눈을 돌려 찾아보면 GNU/Octave, Scilab, Tela, gnuplot, Scigraphica 등과 같이 리눅스에서 사용할 수 있는 수많은 공학용 프로그램들을 찾아볼 수 있다. '작은 것이 아름답다'는 UNIX의 철학을 가장 잘 보여주는 프로그램의 하나인 그래프 도구, gnuplot에 대해 간단히 살펴보고자 한다.

2. 감사의 글

이 문서를 작성하는 데 도움을 주신 많은 분들께 감사드립니다.

- 이 문서를 작성하도록 격려해주신 강차훈, 정춘호님께 감사드립니다. 두 분의 도움이 없었다면 이 문서를 완성되지 못했을 것입니다.
- Curve Fitting에 관한 예제를 번역하도록 허락해 주신 Duke Univ.의 Henri P. Gavin 교수님께도 감사의 말씀을 드립니다.

1장. gnuplot의 소개

1.1. gnuplot의 설치

gnuplot의 공식 홈페이지는 http://www.gnuplot.info이며 Linux/UNIX 이외의 다양한 플랫폼에서 동작하는 바이너리를 구할 수 있다. gnuplot 공식홈페이지 ftp site에서 gnuplot의 공식매뉴얼을 다운받아볼 수 있는데 자그마치 그 내용이 128페이지에 육박한다. 앞에서도 말했듯이 2메가가 채 안되는 적은 용량이라도 그 기능과 빠른 속도에 입을 다물지 못한다. 티코의 겉모습을 한 벤츠라고나 할까?

1.2. gnuplot의 시작

Linux와 UNIX의 경우, 터미널 상에서

```
$ gnuplot
```

이라고 입력하면 그림 1과 같이 gnuplot이 실행된다. 처음 리눅스를 깔았을 때 프롬프트 하나 달랑 뜨는 그 황당함을 기억하는가? 그러나 하나하나 gnuplot의 기능들을 익혀나가다 보면 그 끝이 어디인지를 가늠하기 어렵다.

그림 **1-1.** 그림의 제목

```
[kaien@juliet kaien]$ gnuplot

        G N U P L O T
        Linux version 3.7
        patchlevel 1
        last modified Fri Oct 22 18:00:00 BST 1999

        Copyright(C) 1986 - 1993, 1998, 1999
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual
        The gnuplot FAQ is available from
        <http://www.ucc.ie/gnuplot/gnuplot-faq.html>

        Send comments and requests for help to <info-gnuplot
        Send bugs, suggestions and mods to <bug-gnuplot@dartm


Terminal type set to 'x11'
gnuplot> 



[영어][완성][두벌식]
```

gnuplot의 시작

1.3. gnuplot의 종료

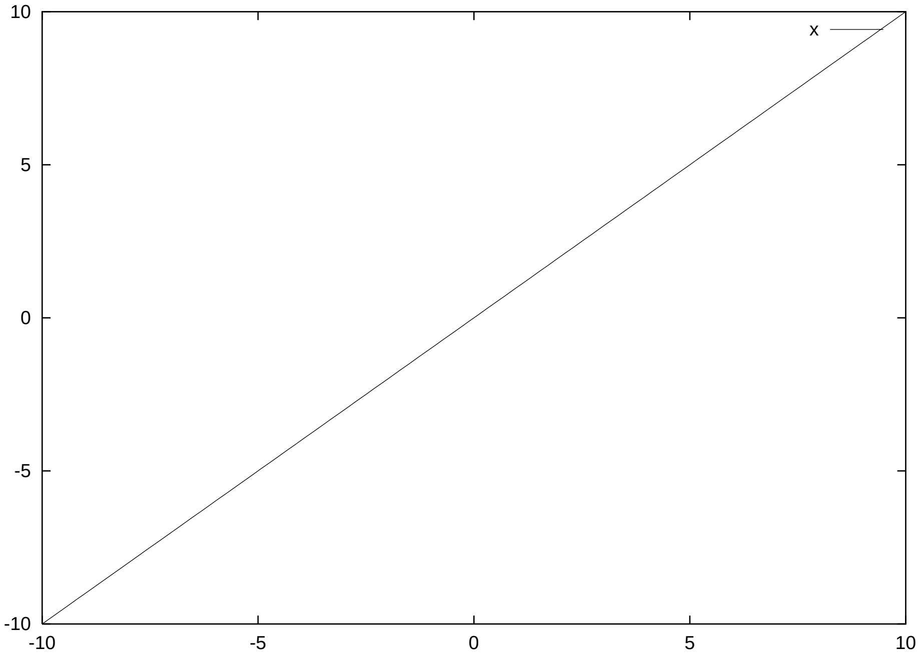
gnuplot의 프롬프트 상에서 q, quit, exit 등의 명령으로 종료한다.

2장. 그래프 그리기

일단 가장 간단한 형태의 2차원 그래프부터 설명한다. gnuplot 프롬프트 상에서

```
gnuplot> plot x
```

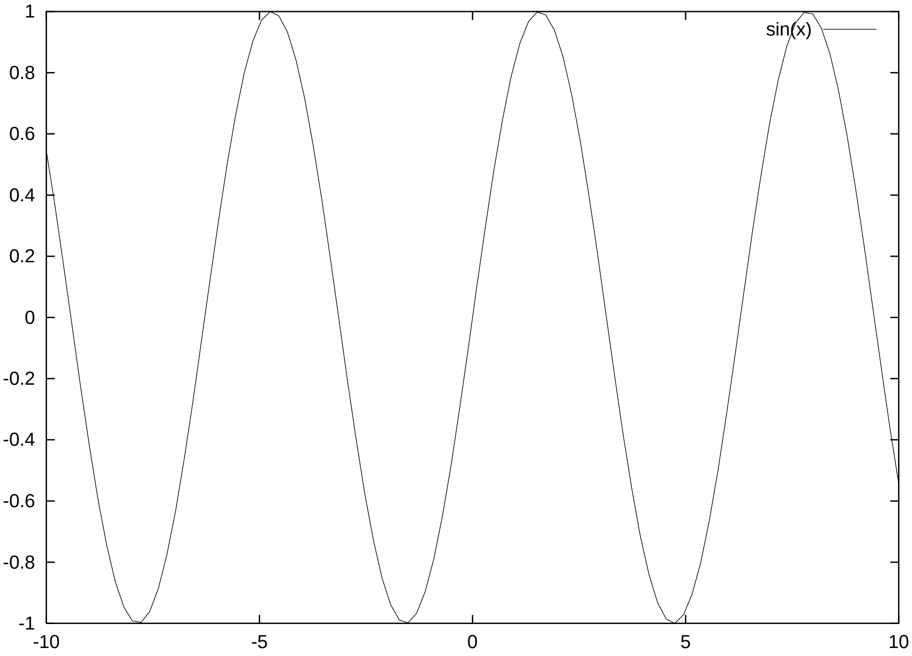
그림 **2-1. y=x**의 그래프



라고 입력하면 그림 2와 같이 그래프가 표시된다. 즉, y=x의 그래프를 나타내는 것이다. 동일한 방법으로

```
gnuplot> sin(x)
```

그림 **2-2. y=sin(x)**



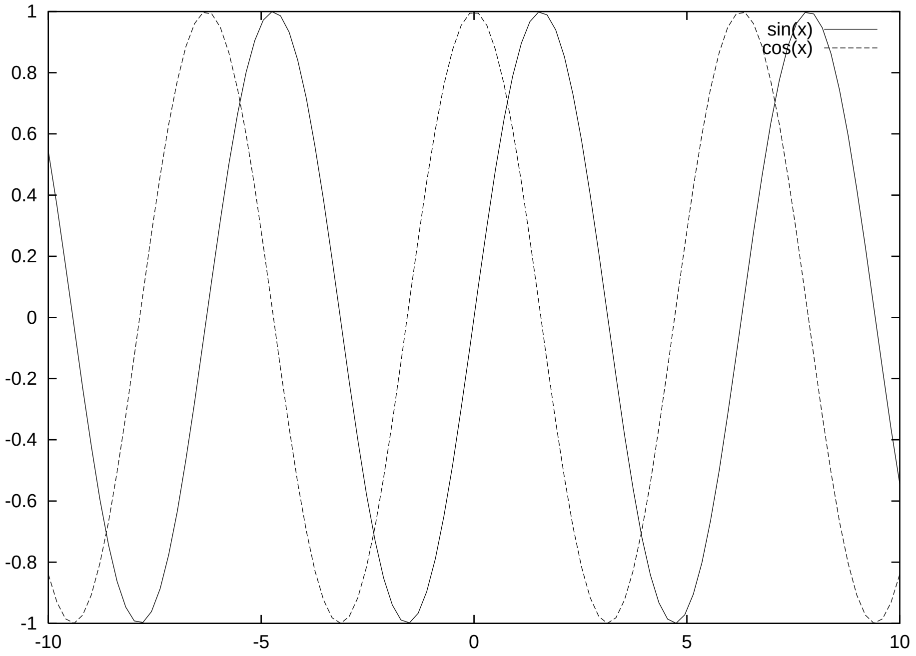
라고 입력하면 위와 같이 출력된다. 두개 이상의 그래프를 동시에 표현하고 싶을 경우에는 replot을 사용하면 된다. 방금 그려진 y=sin(x)의 그래프에 y=cos(x)의 그래프를 추가하려면

gnuplot> replot cos(x)

라고 입력하면 sin(x)의 그래프 위에 cos(x)의 그래프가 겹쳐서 나타나게 된다. replot을 사용하지 않고 ,(comma)를 사용하여 각 그래프를 구분해 주어도 동시에 여러 함수를 한 그래프에 나타낼 수 있다.

gnuplot> plot cos(x),sin(x)

그림 **2-3. y=sin(x),cos(x)**



1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-10
-5
0
5
10
sin(x)
cos(x)

3장. 데이터의 플롯

3.1. 데이터의 종류

실험 등으로 얻은 데이터를 gnuplot에서 그래프로 나태내기 위해서는 적절한 데이터 파일을 만들어주어야 한다.

```
#text1.txt
#x              y
0.0123          5.21
0.168           6.02
0.184           6.08
0.190           6.51
0.226           6.75
0.259           6.81
0.304           7.15
```

다음과 같은 2차원 그래프를 플롯해보자. 여기서 '#' 뒷부분은 주석으로 처리된다. 일반적으로 플롯을 위한 데이터 파일은 ASCII 텍스트 파일이어야 함을 명심하자.

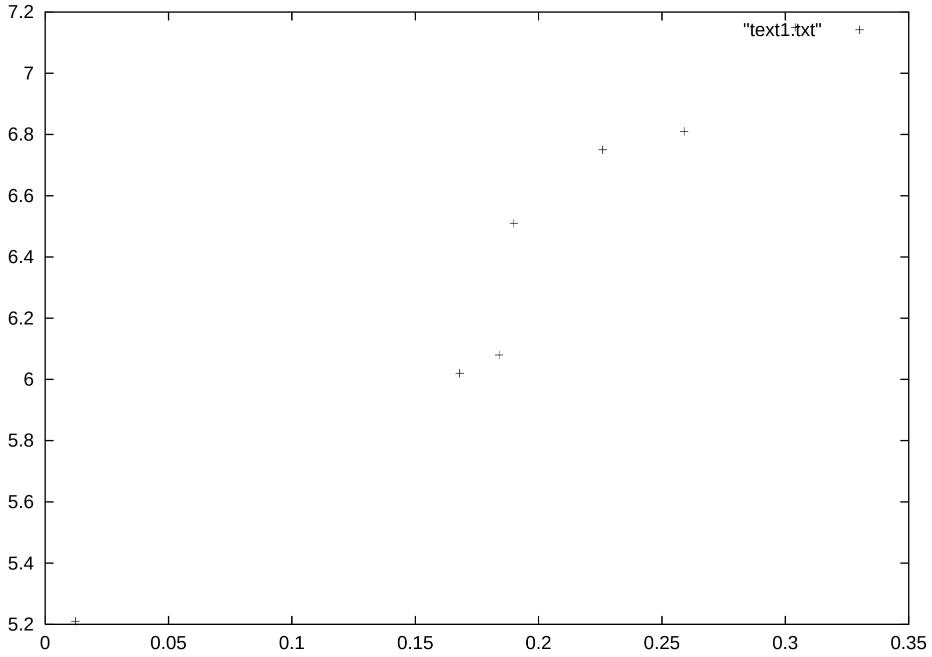
3.2. 데이터의 플롯

위에서 보았던 text1.txt 파일을 플롯하는 경우를 예로 들어보도록 한다. gnuplot 프롬프트 상에서

```
gnuplot> plot "text1.txt"
```

라고 입력하면

그림 **3-1. text1.txt**의 그래프



와 같은 그래프를 얻을 수 있다. gnuplot에서 플롯하고 싶은 데이터가 다른 곳에 위치해 있을 경우 프롬프트 상에서 그 파일의 경로를 다음과 같이 직접 지정해 주면 된다.

```
gnuplot> plot "/home/kaien/data/text1.txt"
```

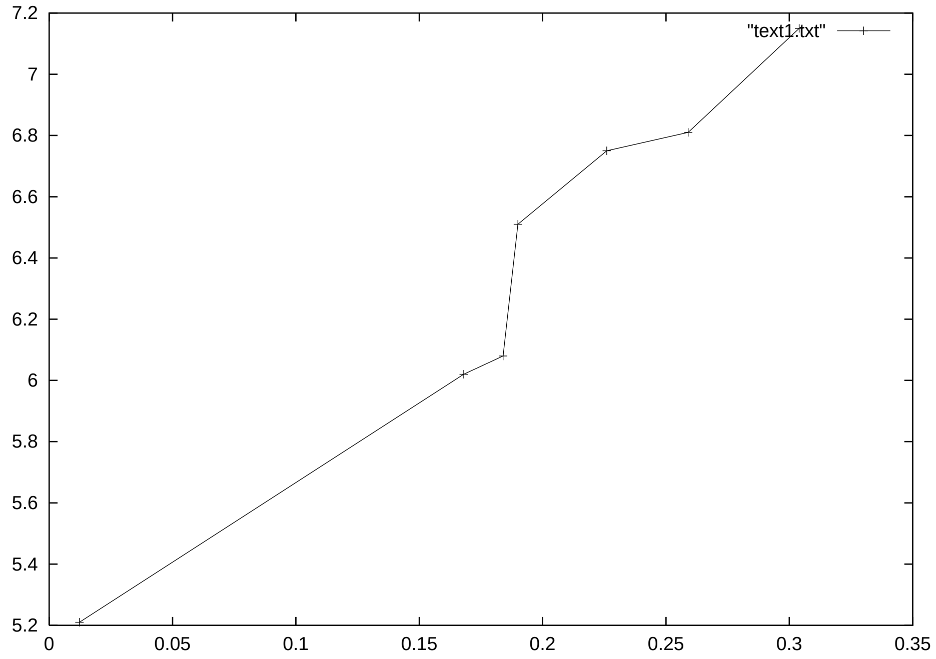
또한 gnuplot은 기본적으로 UNIX tool이므로 'cd' 등의 shell 명령을 모두 사용할 수 있으며 history 기능도 동작하므로 한번 입력했던 그래프를 다시 그리기 위해 똑같은 명령을 다시 입력하지 않고 위,아래 화살표키를 이용하여 조금 전에 입력했던 명령들을 다시 호출할 수 있다.

3.3. 그래프의 형태

그림 5에서 나타난 그래프는 단순한 점으로 나타내어진다. 이 데이터를 하나의 이어진 선으로 표시하고 싶을 경우에는

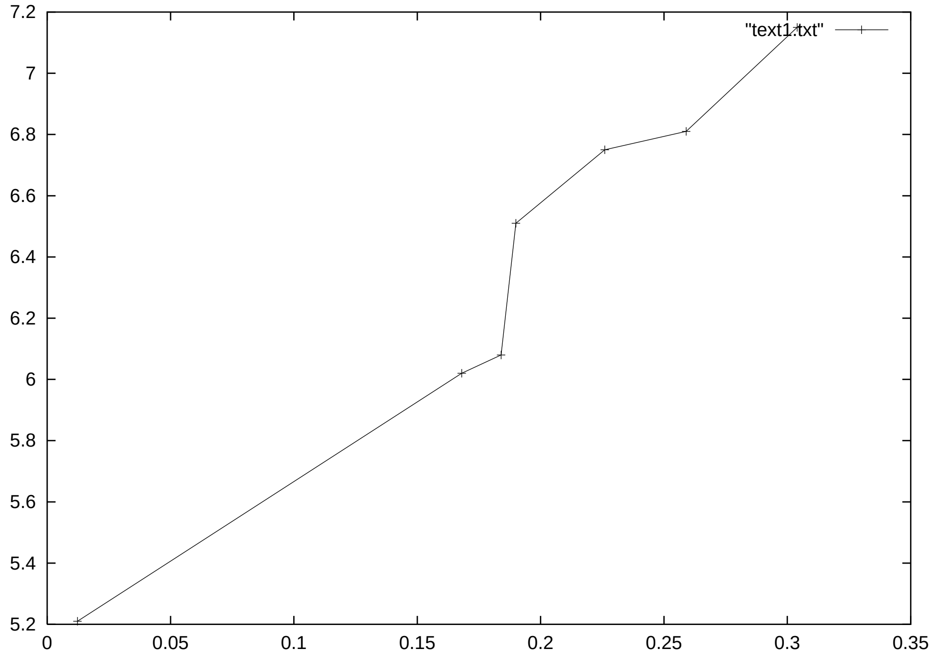
```
gnuplot> plot "text1.txt" with l(ine)
```

그림 **3-2.** 선으로 각 점들을 연결



으로 표시하면 된다. 여기서 'line'이라고 일일이 입력하지 않고 'l'만 입력하여도 같은 결과를 얻을 수 있다. 또, 데이터의 점들을 표시하고 각각의 선으로 잇는 경우에는

그림 **3-3.** 점들을 표시하고 각각의 선으로 잇는다



gnuplot> plot "text1.txt" with linespoints

라고 입력한다.

3.4. 여러 개의 변수가 있을 경우

아래와 같은 경우를 생각해 보자.

```
#text2.txt
#t      x       y        v       E
0.11    0.11    99.94   1       -1.078
0.12    0.12    99.92   1       -1.176
0.13    0.13    99.91   1       -1.274
0.14    0.14    99.90   1       -1.372
0.15    0.15    99.88   1       -1.470
0.16    0.16    99.87   1       -1.568
0.17    0.17    99.85   1       -1.666
0.18    0.18    99.84   1       -1.764
0.19    0.19    99.82   1       -1.862
```

여기서 단순히

gnuplot> plot "text2.txt"

라고 할 경우, 첫 번째와 두 번째의 데이터를 2차원으로 플롯하게 된다. 만일 두 번째, 세 번째 데이터를 플롯하고 싶다면 다음과 같이 입력한다.

```
gnuplot> plot "text2.txt" u(sing) 2:3
```

이와 같이 여러개의 변수로부터 플롯하고자 하는 값을 선택하는 경우에는 'using' 혹은 'u' 뒤에 사용하고 싶은 변수의 열을 ':'을 이용해 지정해 주어야 한다.

4장. 3차원 플롯

4.1. 함수의 플롯

3차원의 그래프를 플롯하기 위해서는 ’splot’을 사용한다.

gnuplot> splot ’expression’

expression에 출력하고 싶은 함수 z=f(x,y)를 입력한다. 예를 들어 다음과 같이 입력하면

gnuplot> splot exp(x)+exp(y)

그림 **4-1. exp(x)+exp(y)**

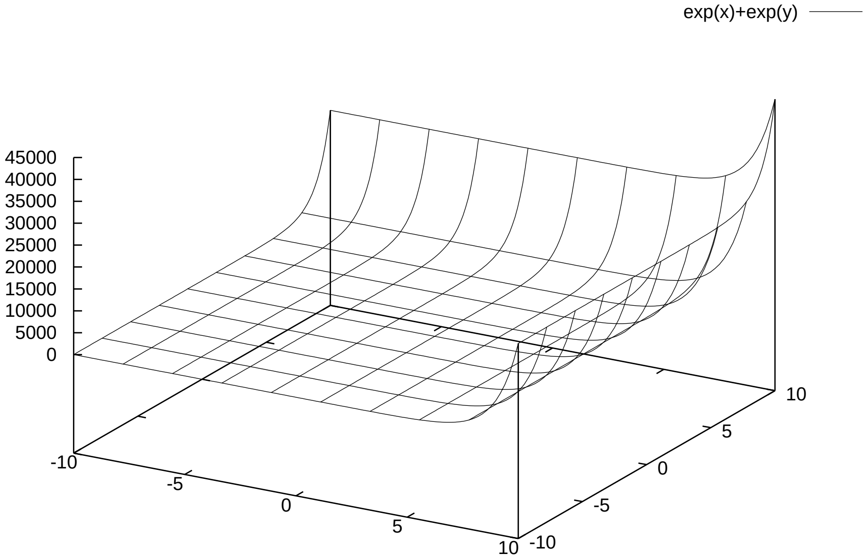


그림 8과 같은 그래프를 얻을 수 있다.

4.2. 데이터의 플롯

3차원에서 데이터를 플롯하는 경우에는 splot 명령을 사용한다. 2차원 함수의 경우와 동일하게

gnuplot> splot "data file"

혹은 다음과 같이 입력한다.

gnuplot> splot 'data file'

4.3. noparametric 플롯 : z값만을 가지는 3차원 데이터 플롯

먼저, noparametric 모드로 전환하기 위해서는 프롬프트 상에서 다음과 같이 입력한다.

gnuplot> set noparatmetric

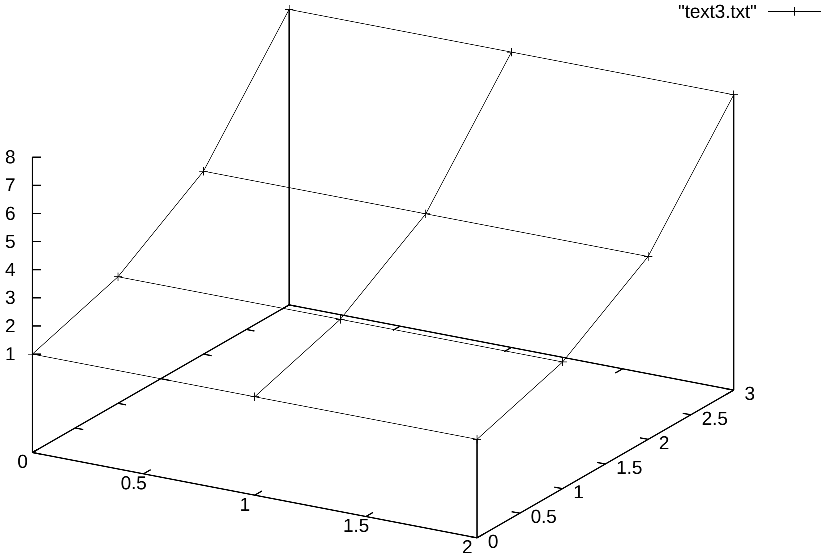
이제 noparametric 모드로 전환되었다. 기본설정이 이 모드이므로 특별히 지정해 줄 필요는 없다. 다음과 같은 데이터 파일이 있다고 하자.

```
#text3.txt
1
1
1

2
2
2

4
4
4

8
8
8
```

이제 이 데이터파일을 플롯해 보자.

gnuplot> splot "text3.txt" with linespoints

그림 **4-2. noparametric** 모드



x, y의 값이 자동적으로 주어지고 그림 9와 같이 표시된다.

4.4. parametric 플롯 : x, y, z의 모든 값을 가지는 3차원 데이터의 플롯

noparametric 모드에서 parametric 모드를 사용하기 위해 프롬프트 상에서 다음의 명령을 입력한다.

```
gnuplot> set parametric
```

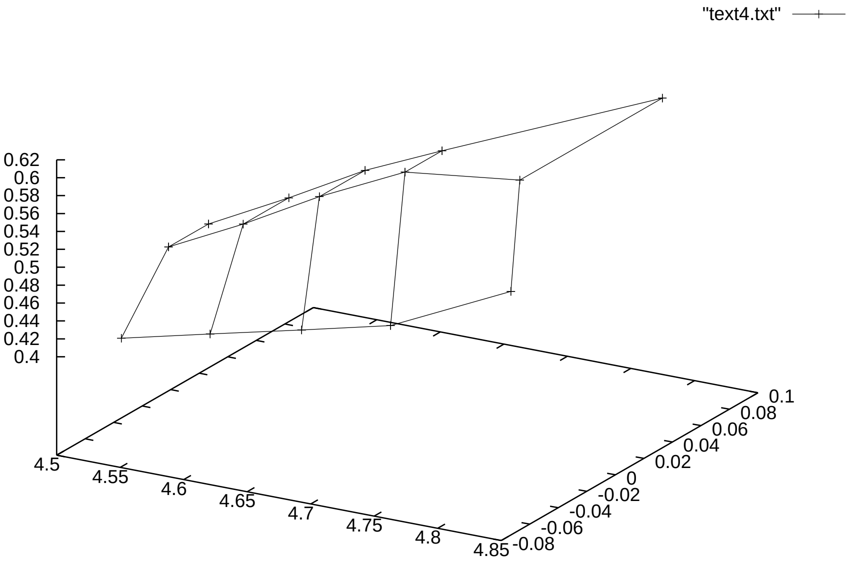
이제 parametric 모드로 전환되었다. 새로운 예를 살펴보자.

```
#text4.txt
#x      y       z
4.514   0.014   0.466
4.575   0.016   0.510
4.635   0.016   0.557
4.699   0.013   0.599
4.786   0.09    0.611

4.514   -0.014  0.466
4.575   -0.016  0.510
4.635   -0.016  0.557
4.699   -0.013  0.599
```

```
4.786   -0.010  0.611

4.533   -0.064  0.415
4.604   -0.065  0.440
4.676   -0.065  0.464
4.746   -0.065  0.488
4.816   -0.043  0.525
```

이 데이터를 플롯하면 다음과 같은 그래프를 얻을 수 있다.

그림 **4-3. parametric** 모드



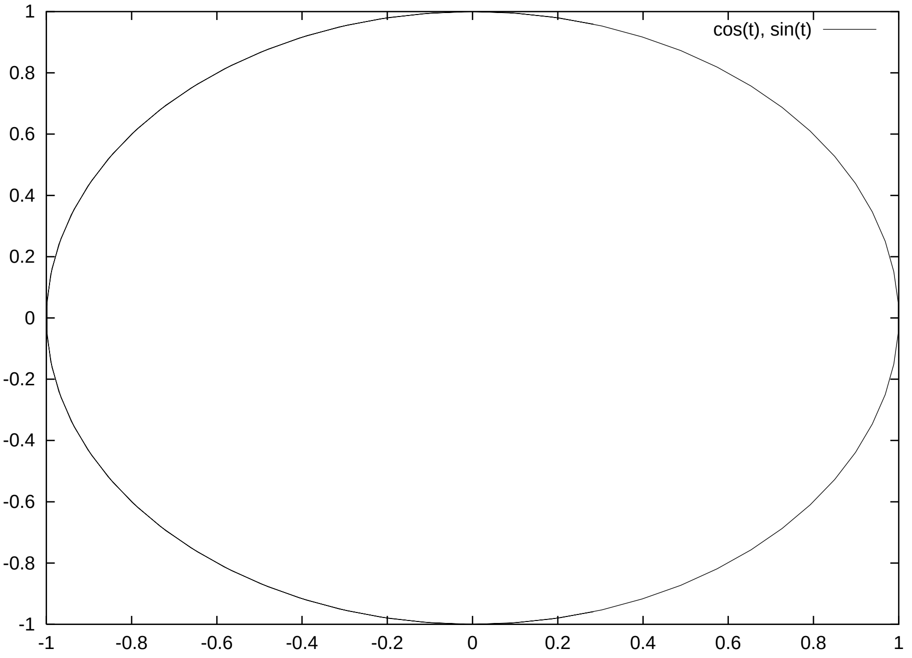
```
gnuplot> splot "text4.txt" with linespoints
```

5장. parameter를 이용한 함수의 플롯

원의 방정식은 매개변수를 이용해 표현하면

```
x=cos(t)
y=sin(t)
```

그림 **5-1.** 실제로는 원의 그래프



로 나타낼 수 있다. 이와 같이 함수를 이용해 표현된 함수의 그래프의 플롯을 생각해 보자. 먼저 gnuplot을 parametrice 모드를 설정한다.

```
gnuplot> set parametric
dummy variable is t for curves, u/v for surfaces
```

이제 원의 함수를 플롯해 보자.

gnuplot> plot cos(t),sin(t)

그래프가 실제 원처럼 동그랗게 보이지 않는 이유는 x, y축의 길이차이 때문이다. 3차원의 플롯도 같은 방식으로 이루어진다. 구의 방정식을 매개변수를 통해 표현하면

```
x=cos(u)cos(v)
y=cos(u)sin(v)
```

```
z=sin(u)
```

플롯해 보면

gnuplot> splot cos(u)*cos(v),cos(u)*sin(v),sin(u)

그림 **5-2.** 구의 그래프

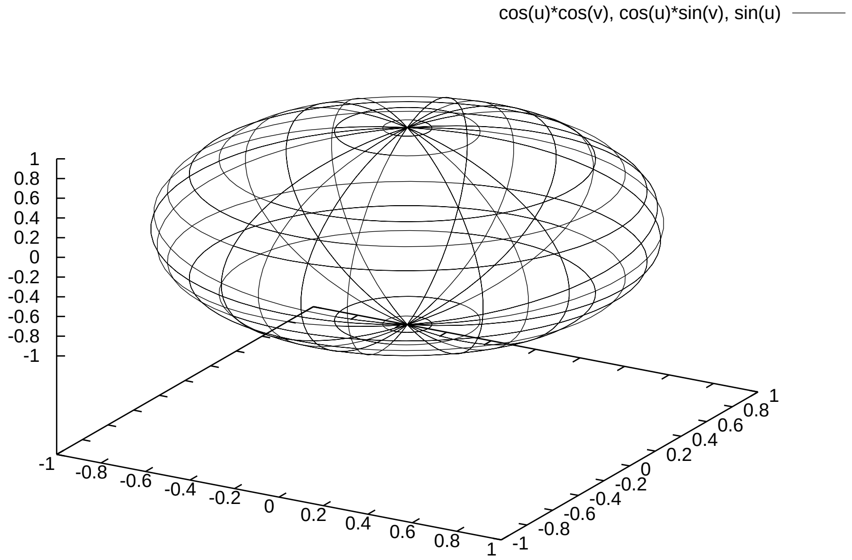


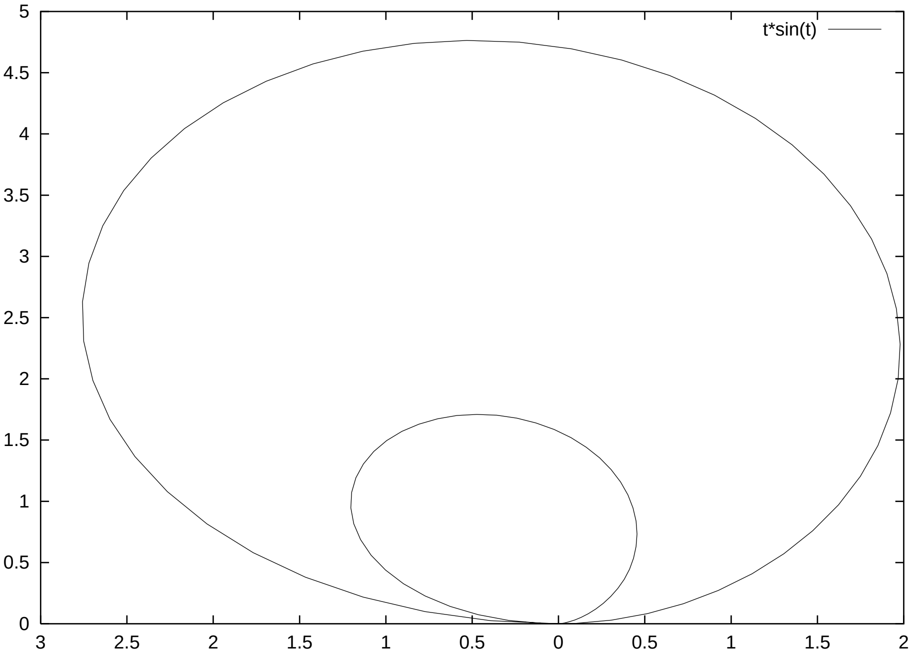
그림 12와 같은 그래프를 얻을 수 있다. 3차원의 parameter는 u,v가 된다.

6장. 극좌표의 플롯

2차원 좌표에 한해서만 극좌표의 플롯도 가능하다. 먼저 다음의 명령으로 극좌표 모드로 변환한다.

gnuplot> set polar

그림 **6-1.** 극좌표 그래프의 예제



이제 극좌표의 그래프를 출력할 수 있다. 극좌표는 r=f(x)의 형태로 나타낸다. r은 원점으로부터의 거리를, x는 회전각을 나타낸다. 회전각은 다음의 명령으로 바꿀수 있다.

```
set angles radians : 각도를 'radian'으로 표현
set angles degrees : 각도를 'degree'로 표현
```

예를 들어

gnuplot> plot t*sin(t)

라고 입력하면 다음과 같은 그래프를 얻을 수 있다.

일반적인 직교 좌표계로 다시 돌릴때는 다음과 같은 명령을 사용한다.

gnuplot> set nopolar

7장. 그래프의 세부설정

그래프의 범위, 눈금 설정, 제목의 표시 등 그래프의 자세한 설정에 대해서 알아본다. 앞에서부터 계속 언급되어 왔지만 gnuplot에서 설정은 set 명령을 통해 이루어진다.

```
gnuplot> set ’option’
```

현재 설정되어 있는 환경을 보기 위해서는 다음과 같이 입력한다.

```
gnuplot> show ’option’
```

또 설정되어 있는 모든 환경을 보고 싶을 경우

```
gnuplot> show all
```

를 입력하면 된다. 아무 것도 설정되어 있지 않은 상태에서도 default 설정이 표시된다. 이러한 set 명령의 사용법은 다른 UNIX tool들이 그러하듯이 help명령을 통해 자세히 알아볼 수 있다.

```
gnuplot> help set
```

gnuplot의 help는 interactive하게 되어 있어 필요한 메뉴의 설명을 계단식으로 찾아볼 수 있다. 예를 들어 ’help set’ 명령을 내리고 나면 하부토픽인 ’term’에 관한 내용을 다시 찾아볼 수 있다. 이,공학에 대한 기본적인 지식만 있다면 이 ’help’ 명령만으로도 필요한 옵션을 모두 찾을 수 있을 정도로 정리가 잘 되어 있으므로 필요할 때마다 ’help’ 명령을 유용하게 잘 사용한다면 큰 도움이 될 것이다.

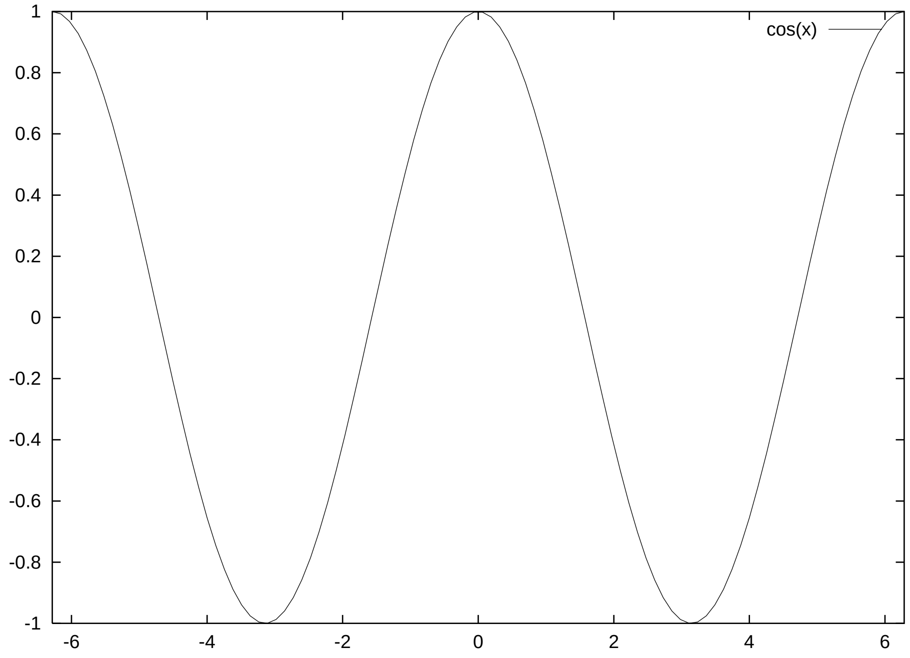
7.1. 축의 범위를 설정

축의 범위를 설정하는 것은 set 명령을 이용하는 것 이외에도 몇 가지 방법이 있다. 첫 번째로

```
gnuplot> plot [-2*pi:2*pi]cos(x)
```

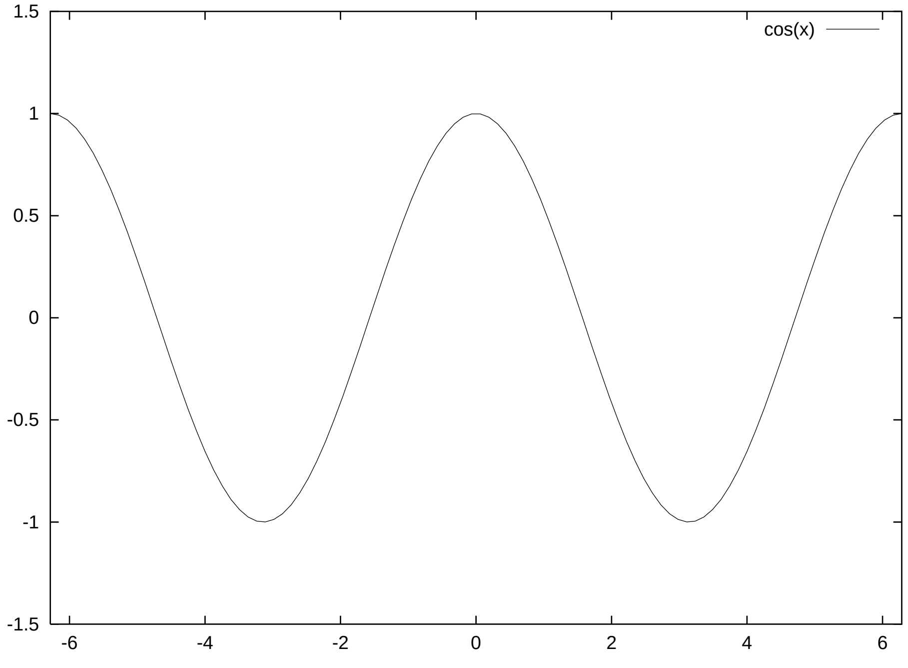
라고 입력해 보자. x축의 범위가 -2π에서 2π가 된다. gnuplot에서 ’pi’는 원주율 π를 의미한다.

그림 **7-1. x**축의 범위를 지정



또 다음과 같은 방법으로 y축의 범위도 지정할 수 있다.

그림 **7-2. x,y**축의 범위를 지정



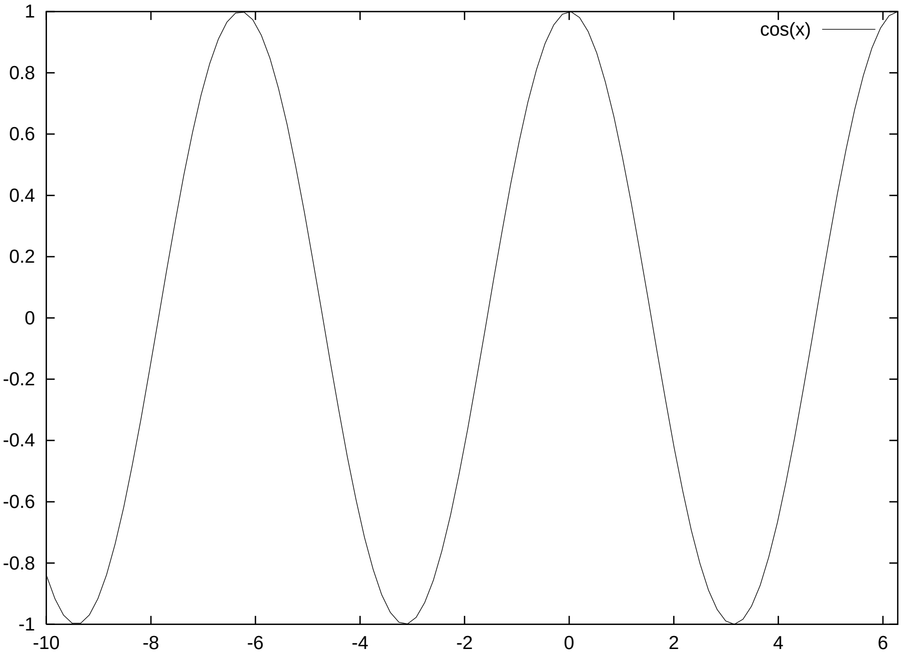
gnuplot> plot [-2*pi:2*pi][-1.5:1.5]cos(x)

즉, 'plot' 뒷부분에 나오는 함수, 데이터파일의 앞 그래프의 범위를 지정할 수 있다. 이것은 3차원의 경우에도 동일하게 적용된다. 또 다음과 같이 지정 y축의 범위만을 지정할 수도 있다.

gnuplot> plot [][-1.5:1.5]cos(x)

지정하지 않는 축의 범위는 임의의 값을 취하게 된다. 상한값이나 하한만을 지정할 수도 있다.

그림 **7-3.** **x**축의 상한값만을 지정



gnuplot> plot [:2*pi]cos(x)

또 'set' 명령을 사용해

```
set xrange[min:max]
set yrange[min:max]
set zrange[min:max]
```

로도 같은 결과를 나타난다. 당연히 min, max에는 각각 최소, 최대값을 입력한다. [:]안에는 위에서 설명한 것과 같이 설정할 수 있다. gnuplot에서는 한번 설정된 값이 이후의 그래프에 계속 영향을 미치게 된다. 원래의 기본값으로 되돌리고 싶을 때에는

gnuplot> set autoscale 'axes'

axes에는 범위를 지정하는 좌표축을 지정하면 된다. 지정할 수 있는 'axes'는 x, y, z, xy의 네 종류가 있다. 모든 축을 되돌리고 싶을 때에는

gnuplot> set autoscale

이라고 입력하면 된다.

7.2. 축의 눈금을 변경

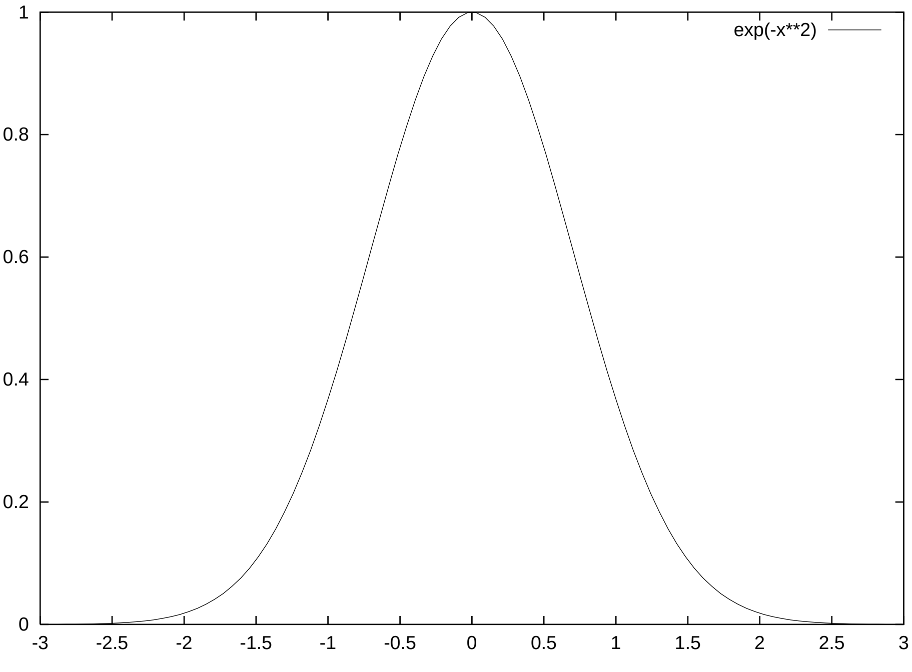
눈금의 폭을 변경하는 방법을 알아보자. 눈금을 설정하기 위해서는 set의 옵션 중에서 xtics(혹은 ytics, ztics)를 사용한다.

gnuplot> set xtic start, incr, end

여기서 start는 눈금의 시작값, incr는 눈금의 폭, end는 눈금이 끝나는 값을 지정한다. 다른 축 ytics, ztics도 동일하게 적용된다. 예를 들어, x축의 눈금을 -3에서 3까지 0.5의 폭으로 새기고 y축의 눈금을 0에서 1까지 0.2의 폭으로 하면서 exp(-x2)의 그래프를 출력하기 위해서는

```
gnuplot> set xtics -3,0.5,3
gnuplot> set ytics 0,0.2,1
gnuplot> plot [-3:3][0:1]exp(-x**2)
```

그림 **7-4.** 눈금의 폭을 설정



또 같은 간격의 눈금나누기 뿐만 아니라, 눈금을 나누는 위치를 자유롭게 지정할 수도 있다.

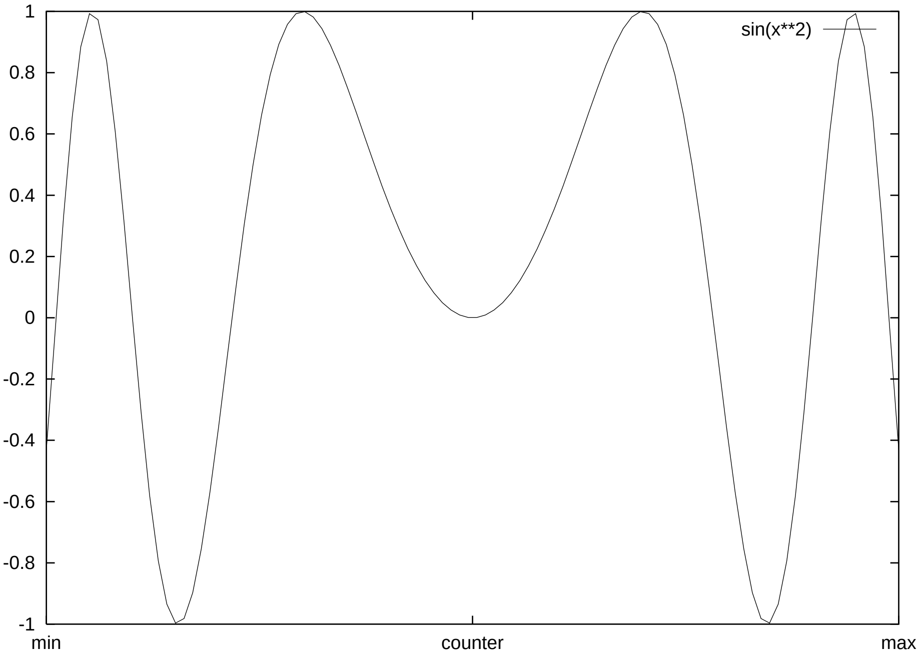
gnuplot> set xtics {position{,position, ...}}

와 같이 지정한다. postion은 붙이고 싶은 눈금의 위치를 지정한다. 예를 들면

```
gnuplot> set xtics(0, pi/8, pi/4, pi/2)
gnuplot> plot [0:pi]sin(x**2)
```

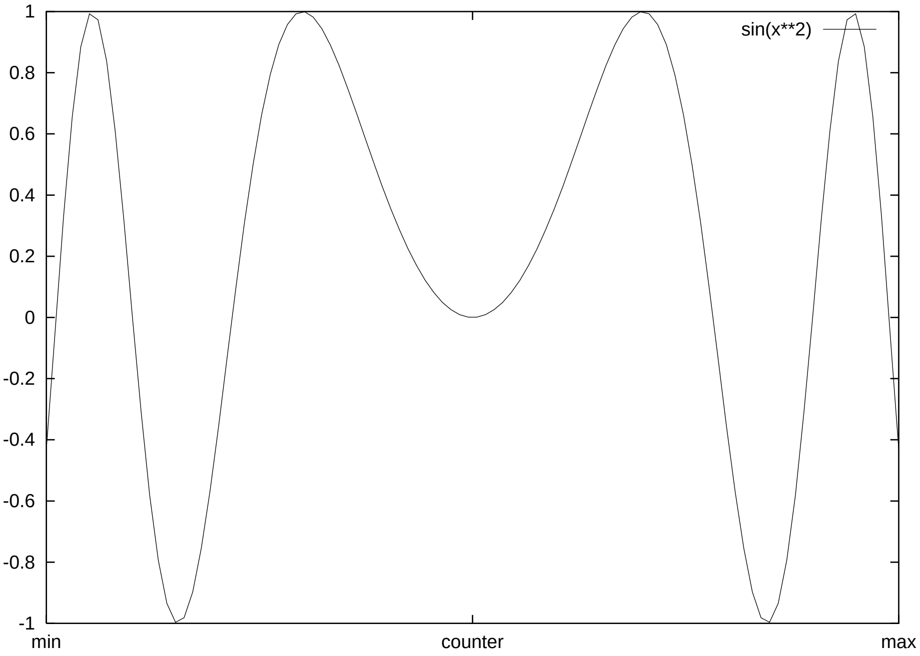
의 그래프는 다음과 같다.

그림 **7-5.** 지정한 위치에 눈금을 표시



여기까지는 눈금에 수치를 지정했지만, 임의의 라벨을 지정할 수도 있다.

그림 **7-6.** 눈금에 라벨을 표시



```
gnuplot> set xtics("min"-pi,"counter"0,"max"pi)
gnuplot> plot [-pi:pi]sin(x**2)
```

설정된 눈금의 값을 기본값으로 되돌릴 때에는 다음과 같이 고친다.

gnuplot> set xtics

여기서 xtics의 경우는 x축, ytics, ztics는 y,z축의 값을 되돌린다. 또 눈금을 원하지 않을 때에는

gnuplot> set notics

라고 해주면 된다.

7.3. 그래프에 제목 표시

대부분의 그래프에는 그래프의 내용을 표시하기 위해 제목을 적어줄 필요가 있다. gnuplot에서 제목을 붙이려면

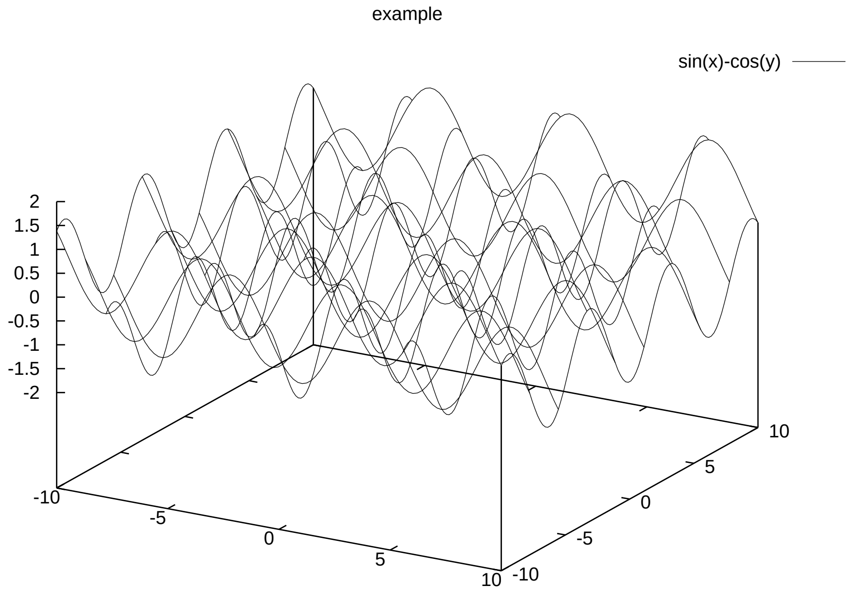
gnuplot> set title "title"

과 같이 설정한다. "title"에는 표시하고 싶은 제목을 ""로 입력한다. 제목은 그래프의 상부중앙에 표시된다.

```
gnuplot> set title "example"
```

```
gnuplot> splot sin(x)-cos(y)
```

그림 **7-7.** 그래프에 제목을 표시



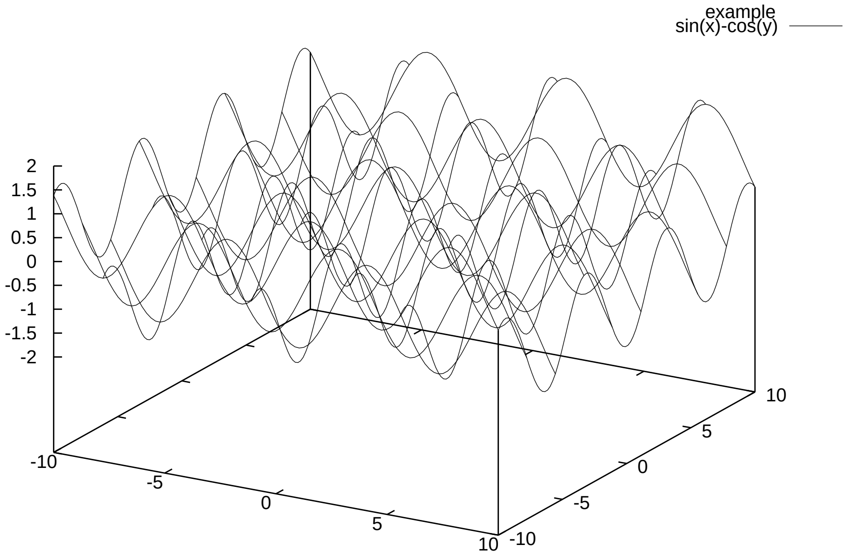
제목의 위치를 지정하고 싶을 때에는 다음과 같이 설정한다.

gnuplot> set title "title" x,y

x,y에는 수치를 대입한다. 오른쪽으로 x, 위쪽으로 y만큼 이동한다.

```
gnuplot> set title "example" 30,-3
gnuplot> splot sin(x)-cos(y)
```

그림 **7-8.** 제목의 위치 이동



더 이상 제목을 표시하고 싶지 않을 때에는

gnuplot> set title

이라고 입력한다.

7.4. 축에 설명 붙이기

각 축에 설명을 위한 라벨을 붙여보자. 축에 라벨을 표시하기 위한 명령은

```
gnuplot> set xlab(el) "label"
gnuplot> set ylab(el) "label"
gnuplot> set zlab(el) "label"
```

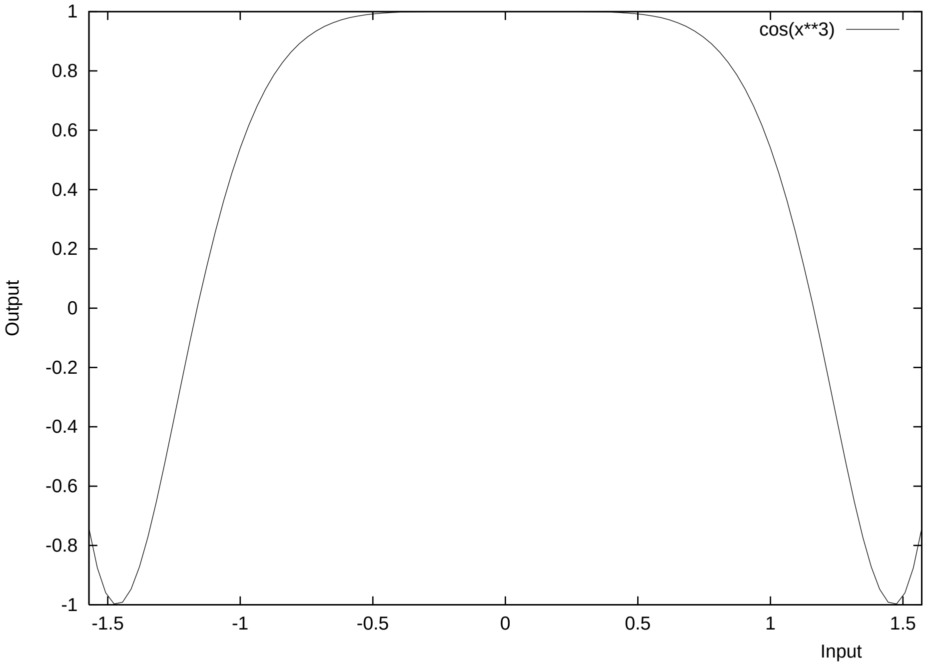
이 label에 지정된 문자가 축의 라벨로 표시된다. 라벨의 표시위치는 2차원의 경우 x축의 중앙과 y축의 좌측 위에, 3차원에서는 x축의 중앙, y축의 중앙, z축의 상부가 된다. 제목과 같이 "label" 뒤에

gnuplot> set xlab(el) "label" x,y

x,y에 수치를 넣어 오른쪽으로 x, 위쪽으로 y만큼 이동한다.

```
gnuplot> set xlab "Input" 30
gnuplot> set ylab "Output"
gnuplot> plot [-pi/2:pi/2]cos(x**3)
```

그림 **7-9. x,y**축에 라벨을 표시



라벨을 표시하지 않기 위해서는

gnuplot> set xlabel

이라고 한다.

7.5. 곡선에 이름 붙이기

지금까지 여러 예에서 오른쪽 위에 곡선의 방정식, 데이터 파일들의 파일명 등이 표시되었다. 이 그래프의 설명을 gnuplot에서는 'key'라고 한다. Key는 표현된 곡선의 그려진 차례대로 윗부분에 표시된다. 이 key를 자유롭게 변경할 수 있는 방법을 알아보자.

gnuplot> plot expression title "curve_name"

그래프의 방정식 데이터 파일, 혹은 expression으로 "curve_name"이 화면의 오른쪽 상단에 표시된다.

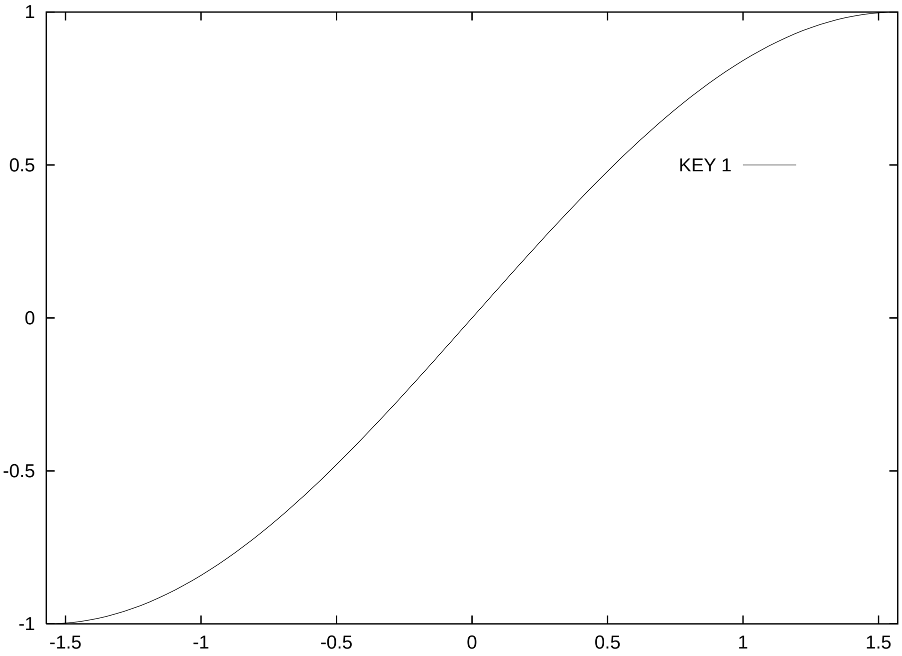
gnuplot> plot [-pi/2:pi/2]sin(x) title "KEY 1"

그러나 이 상태에서는 key와 그래프가 겹쳐서 보기가 그다지 좋지 않다. 여기서

```
gnuplot> set key 1,0.5
gnuplot> plot [-pi/2:pi/2]sin(x) title "KEY 1"
```

와 같이 key를 적절한 곳으로 옮겨주면 훨씬 더 보기가 좋아진다.

그림 **7-10. KEY**의 적절한 이동



삼차원의 경우에는

gnuplot> set key x,y,z

로 표시해주면 (x,y,z)의 위치에 key가 표시된다. 또 key를 표시하지 않는 경우는 다음과 같이 해주면 된다.

gnuplot> set nokey

7.6. 좌표축의 표시

2차원의 그래프를 플롯했을때, 그래프 영역에 원점이 포함되어 있는 경우 그래프에 좌표축이 그려질 때 이를 제어하는 방법을 알아보자.

```
set zeroaxis  : x, y축을 점선으로 표시
set nozeroaxis : x, y축을 표시하지 않음
set xzeroaxis : x축을 점선으로 표시
set xnozeroaxis : x축을 표시하지 않음
```

이 명령은 2차원의 경우에도 사용할 수 있는 명령이므로, 좌표축을 설정한 후에 3차원 그래프를 그려도 아무 영향을 미치지 못한다.

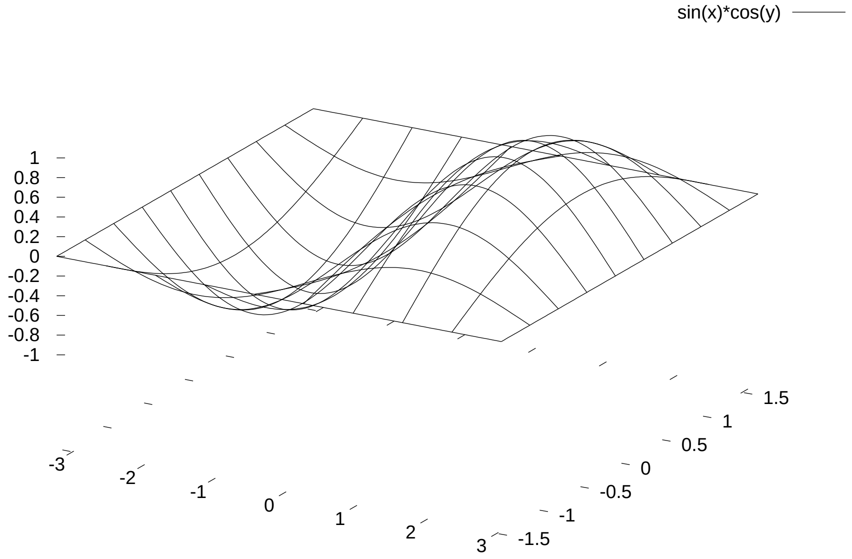
7.7. 경계의 표시

지금까지의 모든 2차원 그래프에는 눈금 위치에 눈금을 나눈 경계가 표시되어 있다. 3차원일 경우 x-y평면과 z축의 경계와 함께 표시되어 있다. 다음 명령을 사용해 이 경계의 표시 여부를 설정한다.

```
set (no)border : 그래프의 경계를 표시(해제)
```

```
gnuplot> set noborder
gnuplot> splot [-pi:pi][-pi/2:[pi/2]sin(x)*cos(y)
```

그림 **7-11.** 경계의 해제



경계를 표시하지 않도록 설정해도 눈금은 그대로 남아있으므로, 이 명령을 사용할 때에는 눈금도 같이 없애는 것이 바람직하다.

7.8. 격자의 표시

그래프를 좀더 쉽게 알아보기 위해 그래프에 격자를 표시하고 싶을 경우 ’grid’ 명령을 이용한다

```
set (no)grid   : 격자의 설정(해제)
```

7.9. log 스케일의 표시

gnuplot에서는 log 그래프도 나타낼 수 있다. 앞에서도 보아왔지만 삼각함수, 지수함수, log 함수 등 대부분의 함수를 지원한다.

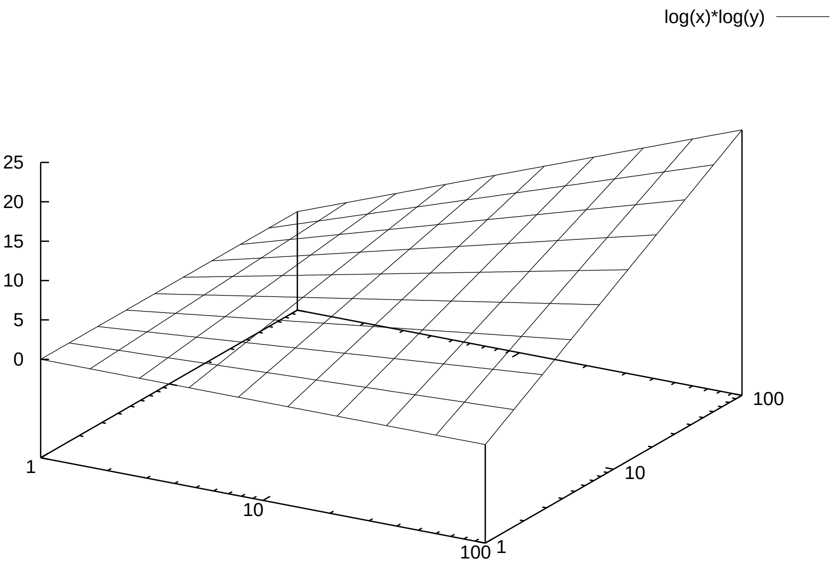
gnuplot> set logscale ’axes’

’axes’에 logscale의 축을 지정한다. 축의 지정에는 x, y, z 및 모든 대칭되는 축을 지정할 수 있다. 축의 지정을 생략한 경우 모든 축이 log 스케일이 된다. 축을 원래의 스케일로 돌릴 경우 다음의 명령을 사용한다.

gnuplot> set nologscale ’axes’

```
gnuplot> set logscale xy
gnuplot> splot [1:100][1:100]log(x)*log(y)
```

그림 **7-12.** 축을 **log** 스케일로 표시



logscale 명령과 같이 'axes'에는 x, y, z 및 모든 대칭축을 지정할 수 있다.

8장. 그래프의 추출

실험 등에 얻어진 데이터를 그래프로 플롯해서 눈으로만 확인하는 경우는 몹시 드물다. 실제 플롯된 데이터에 제목을 표시하고, 각각의 축에 인덱스를 다는 등의작업을 거친후 논문이나 보고서 등에 삽입된다. 이런 그래프를 추출하기 위해서는 gnuplot의 결과를 출력할 터미널을 지정해주어야 한다. 맨 첫머리에 gnuplot을 리눅스 X-Windows상에서 실행할 경우 기본적으로 터미널은 ’x11’로 설정된다. 이외에도 여러 가지 형식으로 그래프를 추출할 수 있다. 기본적인 설정방법을 알아보자.

```
gnuplot> set term(inal) ’type’
gnuplot> set output ’filename’
```

의 형식으로 지정을 한다. 더 자세한 내용은

gnuplot> help set term

으로 참조하기 바란다. 예를 들어 png 파일 형식으로 그래프를 그리고 싶을 때에는

```
gnuplot> set term png
gnuplot> set output ’plot.png’
gnuplot> plot x
```

이런 식으로 입력해 주면 gnuplot을 실행한 위치에서 y=x의 그래프가 ’plot.png’ 파일로 저장된다. 나는 LaTeX 문서나 sgml 문서 등에 그래프를 삽입할 경우를 생각해 가급적 그래프를 eps(Encapsulated Post Script)파일로 저장한다. 다른 형식에 비해 PostScript 파일이 훨씬 더 품위있어 보이고, 깔끔하기 때문에 개인적으로 선호하는 편이다. 앞에서 나온 예들도 모두 eps 파일로 작성된 예제들이다.

```
gnuplot> set term post
gnuplot> set output ’sinx.eps’
gnuplot> plot sin(x)
```

하지만 이렇게 만들어진 eps 파일은 ghostscript를 지원하는 viewer, ghostview와 같은 프로그램에서만 볼 수 있고 웹에 올리거나 일반문서에 포함시키기는 어렵다. jpg나 gif 파일로 바로 추출할 수 있다면 좋겠지만 아쉽게도 라이센스 문제로 3.7 버전부터 제외되었다. 이런 eps 파일을 변환시키기 위해서 ImageMagick의 convert를 이용하면 매우 편리하다. 이 ImageMagick 패키지도 일반적인 배포판에는 모두 설치가 되어 있을 것이다. 이 패키지 안에는 간단한 그래픽 작업을 하는데 필요한 명령들이 대부분 들어 있다(파일변환의 convert, 간단한 그래픽 작업의 display 등). 자세한 내용은 직접 man 페이지나 매뉴얼 등을 참조하기로 하고 여기서는 convert를 소개하고자 한다. ImageMagick의 convert의 기본적인 사용법은 다음과 같다.

$ convert test.jpg test.gif

아주 간단한 명령만으로 jpg파일이 gif 파일로 바뀐다. 너무 허무할 정도로 간단하다. ‘작은 것이 아름답다’라는 UNIX의 철학을 다시 한번 느낄수 있다. 이와 같은 방법으로 eps 파일도

jpg이나 gif, png 등의 다른 파일로 쉽게 변환할 수가 있다. 그러나 convert를 사용, eps 파일을 변환시키면 -90도 회전시켜서 변환을 한다. 그래서 단순히

```
$ convert test.eps test.jpg
```

로 변환하면 그림이 90도 회전되어 있다. 이를 방지하기 위해서

```
$ convert -rotate 90 test.eps test.jpg
```

의 명령으로 깔끔하게 변환해 준다.

9장. 수식, 기호의 표현

일반적으로 이,공학용 데이터를 그래프로 나타낼 경우에는 앞에서 나온 예보다는 좀더 복잡한 형태의 수식이 필요하다. 이런 수식의 입력하는 가장 손쉬운 방법은 크게 LaTeX을 이용하는 방법과 강화된(enhanced) Postscript를 사용하는 두가지 방법이 있다.

9.1. Postscript enhanced 옵션

많은 옵션들을 이용하면 원하는 수식을 그래프 상에 나타낼 수 있다. ?글의 수식편집기를 기억하는가? 이 수식편집기도 원래는 TeX의 수식입력방식과 매우 유사하다. 여기서 쓰일 수식의 입력방식도 이와 비슷함을 알 수 있다. 자세한 내용은 gnuplot 패키지에 포함되어 있는 ps_guide.ps를 참고하기 바란다. 처음에는 다소 불편하게 느껴질 수도 있겠지만 곧 익숙해진다. 마우스로 일일이 클릭해서 넣는 것보다는 훨씬 속도도 빠르다. 입력방식의 예를 몇 가지 들어보면 위 첨자는 '^', 아래첨자는 '_', 특수문자는 PostScript 문자 코드를 참조해서 입력하면 된다(문자코드는 ps_guide.ps 파일의 두 번째 장에 들어있다). 즉,

```
gnuplot> set term post(script) enhance
gnuplot> set output 'eqsample.eps'
gnuplot> set title '{/Symbol=18 \362@_{/=9.6 0}^{/=12 \245}} \
{/Helvetica e^{-{/Symbol m}^2/2} d} \
{/Symbol m = (p/2)^{1/2}}'
gnuplot> plot x
```

라고 해주면 실제 제목은 그림과 같이 나타난다. 여기서 각 행의 끝에 '\'는 줄이 끝나지 않고 이어지고 있음을 나타낸다.

9.2. LaTeX을 이용한 수식의 입력

일반적으로 논문 등을 LaTeX으로 작성할 경우 gnuplot으로 그려진 그래프를 넣는 방법을 알아보자. 터미널은 latex으로 지정해 주면 된다.

```
set terminal latex
set output "graph.tex"
set size 3.5/5, 3/3.
set format y "$%g$"
set format x "$%5.1f\mu$"
set title "This is a title"
set xlabel "This is the $x$ axis"
set ylabel "This is\\a longer\\version\\ of\\the $y$\\ axis"
set label "Data" at -5,-5 right
set arrow from -5,-5 to -3.3,-6.7
set key -4,8
```

```
set xtic -10,5,10
plot [-10:10] [-
10:10] "sample.dat" title "Data File"  with linespoints 1 7,\
     3*exp(-x*x)+1  title "$3e^{-x^{2}}+1$" with lines 4
```

이런 식으로 그래프를 작성하고 latex 파일의 본문에 그림을 넣어야 할 곳에

```
\begin {figure}
      \begin{center}
        \bogosityincarnate{graph}
      \end{center}
\end {figure}
```

와 같은 내용을 삽입해 주면 된다.

9.3. LaTeX에 eps 파일의 삽입

LaTeX 파일에 eps파일을 삽입할 경우에는 epsfig 패키지를 사용하면 된다.

```
 \documentclass{article}

      \usepackage{amsmath}
      \usepackage{epsfig}

      \begin{document}
      This is a 3-d plot.

      \begin{figure}[ht]
      \begin{center}
      \epsfig{file=plot.ps, angle=-90, width=\linewidth}
      \caption{ $z=\cos(x)\sin(y)$ }
      \end{center}
      \end{figure}

\end{document}
```

와 같은 형식으로 eps 파일을 삽입한다.

10장. 파일로부터 읽어들여 보자!

앞에서 살펴본 모든 내용들은 gnuplot 프롬프트 상에서 직접 입력하는 방법이었고, 사실 이 방법은 잘 쓰이지 않지만 gnuplot의 전반적인 기능을 알아보고자 사용하였다. 실제 gnuplot을 사용할 때 gnuplot의 여러 명령들과 설정을 담고 있는 파일을 생성하여 그 파일을 읽어들여 작업을 하게 된다. gnuplot이 읽어들일 작업파일은 파일의 이름은 상관없이 형식만 ASCII 텍스트 파일이면 된다.

```
$ cat test.plot
#test.plot
set term post
set output 'test.eps'
set xlab "Time(hour)"
set ylab "Distance(meter)"
set title "Velocity of My Car"
set nokey
plot "test.dat" with linespoints

$ gnuplot test.plot
$
```

대부분의 리눅스용 프로그램들이 그러하듯이 무소식이 희소식이다. 디렉토리를 살펴보면 test.eps 파일이 생성되어 있는 것을 확인할 수 있을 것이다. 이렇게 파일을 불러들여 실행하게 프롬프트 상에서 직접 입력하는 것보다 훨씬 더 시간을 줄일 수 있게 되고 shell script와의 연동도 쉬워지게 된다. 이제 gnuplot의 사용에 어느 정도 익숙해 졌다고 보고 실제로 사용될 수 있을만한 script를 하나 분석해보자.

10.1. gnuplot와 shell script의 연동

여기 2차원 데이터로 이루어진 몇 개의 실험데이터 파일이 있다. 파일의 이름은 모두 날짜별로 되어있고 약간의 설명이 들어가 있다.

```
$ ls
Fe011102.dat     Al011102.dat     FeAl011102.dat     Cr012502.dat
```

다음과 같은 script로 쉽게 각 데이터 파일의 그래프를 그려볼 수 있을 것이다. 아주 간단한 파일이지만 생각보다 일을 훨씬 간편하게 해준다. Bash shell의 file extension 기능을 사용한 것을 잘 살펴보기 바란다.

```
#!/bin/bash
# Very Simple Script for Plotting 2-D data
for i in *
do
```

```
touch ${i%.dat}.plot
echo set title "${i%.dat}" >> ${i%.dat}.plot
echo set term post >> ${i%.dat}.plot
echo set output \"${i%.dat}.eps\" >> ${i%.dat}.plot
echo set xlabel \"xlabel\" >> ${i%.dat}.plot
echo set ylabel \"ylabel\" >> ${i%.dat}.plot
echo plot \"${i%.dat}.dat\" with l >> ${i%.dat}.plot

echo Making a .plot for ${i%.dat}.dat file..

gnuplot ${i%.dat}.plot

echo Converting eps file to jpg file..

convert -rotate 90 ${i%.dat}.eps ${i%.dat}.jpg
done

mkdir jpg
mkdir eps
mv *.jpg jpg
mv *.eps eps
```

11장. 몇 가지 부가적인 기능들

사실 gnuplot의 기능들을 세세히 설명하려면 엄청난 분량이 된다. 하지만 유용하게 쓰일 수 있을만한 기능 몇가지들만 더 소개를 하고자 한다. 다음의 'force.dat' 파일을 생각해보자.

```
# This file is called   force.dat
# Force-Deflection data for a beam and a bar
# Deflection    Col-Force       Beam-Force
0.000             0               0
0.001           104              51
0.002           202             101
0.003           298             148
0.0031          290             149
0.004           289             201
0.0041          291             209
0.005           310             250
0.010           311             260
0.020           280             240
```

11.1. Spread sheet 기능

마땅한 제목이 없어 Spread sheet 기능이라 적었지만 실제로는 awk에서와 같은 각 컬럼의 데이터에 약간의 계산을 gnuplot 자체적으로도 할 수가 있다. sin(col.3 + col.1) 對 3 * col.2와 같이 어느 정도 계산이 필요한 그래프를 그리고자 할때 gnuplot에서 다음과 같은 명령으로 결과를 바로 얻을 수 있다.

gnuplot> plot 'force.dat' using (3*$2):(sin($3+$1))

11.2. Curve의 fitting

실제 주어진 데이터의 그래프를 그리고, 그 그래프의 함수를 찾아내는 것이 필요할 때가 있다. 그래프의 대략적인 함수를 알고 그 함수를 계수를 찾아내는 방식으로 접근해 나간다. gnuplot에서 이 문제를 해결해 보자.

force.dat 파일의 함수를 f1(x)이라 정의하고 그 계수, a1과 b1을 찾아보자.

```
f1(x) = a1*tanh(x/b1)             # fitting하고자 하는 함수를 정의한다.
a1 = 300; b1 = 0.005;             # a1과 b1의 초기값을 정의한다
fit f1(x) 'force.dat' using 1:2 via a1, b1

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a1      = 308.687                  +/- 10.62         (3.442%)
```

```
b1      = 0.00226668             +/- 0.0002619    (11.55%)
```

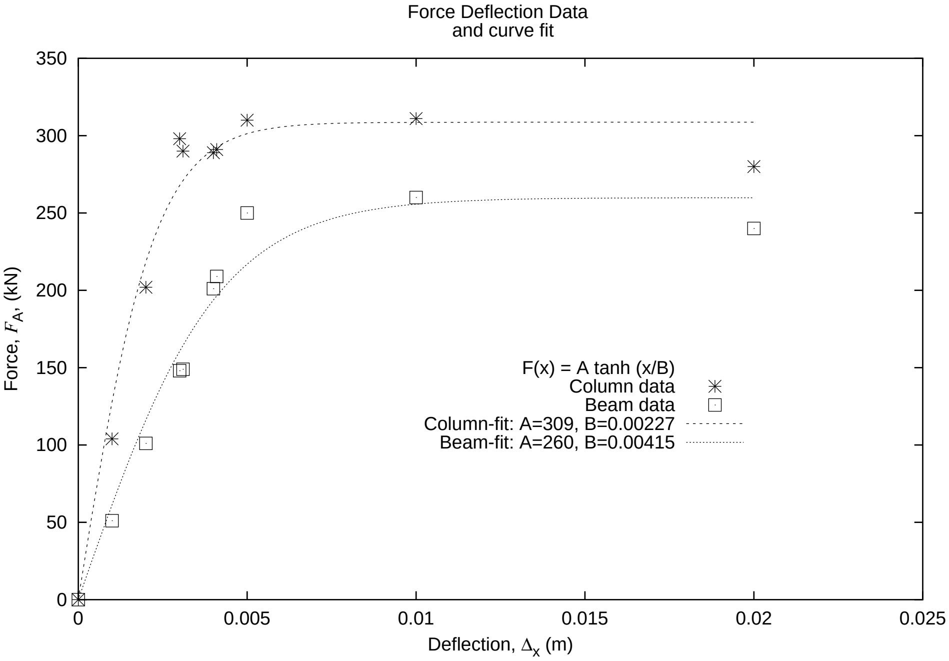
그리고 :

```
f2(x) = a2 * tanh(x/b2)
a2 = 300; b2 = 0.005;
fit f2(x) 'force.dat' using 1:3 via a2, b2

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a2      = 259.891                  +/- 12.82        (4.933%)
b2      = 0.00415497               +/- 0.0004297    (10.34%)
```

이제 fitting 된 값을 이용해서 그래프를 그려보자.

```
set term post enhance
set output 'force.eps'
set key 0.018,150 title "F(x) = A tanh (x/B)"
set title "Force Deflection Data \n and curve fit"      # \n을 이용 제목의 줄
을 바꾼다.
set pointsize 1.5                                # larger point!
set xlabel 'Deflection, {/Symbol D}_x (m)'
set ylabel 'Force, {/Times-Italic F}_A, (kN)'
plot    "force.dat" using 1:2 title 'Column data' with points 3, \
      "force.dat" using 1:3 title 'Beam data'   with points 4, \
      a1 * tanh( x / b1 ) title 'Column-fit: A=309, B=0.00227', \
      a2 * tanh( x / b2 ) title 'Beam-fit: A=260, B=0.00415'
```

그림 **11-1.** 그래프의 **fitting**


Force Deflection Data
and curve fit
Force, F_A, (kN)
Deflection, Δ_x (m)
350
300
250
200
150
100
50
0
0
0.005
0.01
0.015
0.02
0.025
F(x) = A tanh (x/B)
Column data
Beam data
Column-fit: A=309, B=0.00227
Beam-fit: A=260, B=0.00415

12장. 맺으며..

사실 한정된 내용에 방대한 gnuplot의 모든 내용을 다룬다는 것이 처음부터 무리였을지도 모른다. 대부분의 이공학도들의 실험데이터는 텍스트 파일로 쏟아져 나오고, 이러한 방대한 양의 그래프를 그리기 위해 Excel이나 Origin에서 지겨운 반복작업을 되풀이 하는 것을 많이 보아왔다. 이 글을 계기로 이,공학도들이 좀더 리눅스에 관심을 가져주었으면 하는 바램이다. 부족하긴 하지만 gnuplot의 개괄적인 내용은 다루었다고 생각된다. 자세한 내용은 gnuplot의 공식 매뉴얼을 참조하면 더 많은 내용을 얻을 수 있을 것이다.