Creating Graphs in **GAUSSplot**

1 Plot Types

There are four basic plot types in **GAUSSplot**, from which all of the plot types are derived. The basic and extended plot types are as follows:

Basic Plot Types	Extended Plot Types
XY Line Plot	XY Line Plot
	Bar Plot
Polar Line Plot	Polar Line Plot
2D Cartesian Plot	2D Cartesian Plot
	2D Contour Plot
3D Cartesian Plot	3D Cartesian Plot
	3D Contour Plot

2 GAUSSplot Terminology

In **GAUSSplot**, a graph is created in a **workspace**, which is the page on which the graph is drawn. The most basic unit in a **workspace** is a **frame**. A graph can contain multiple **frames**, each of which contain a set of axes (although you may choose not to display them).

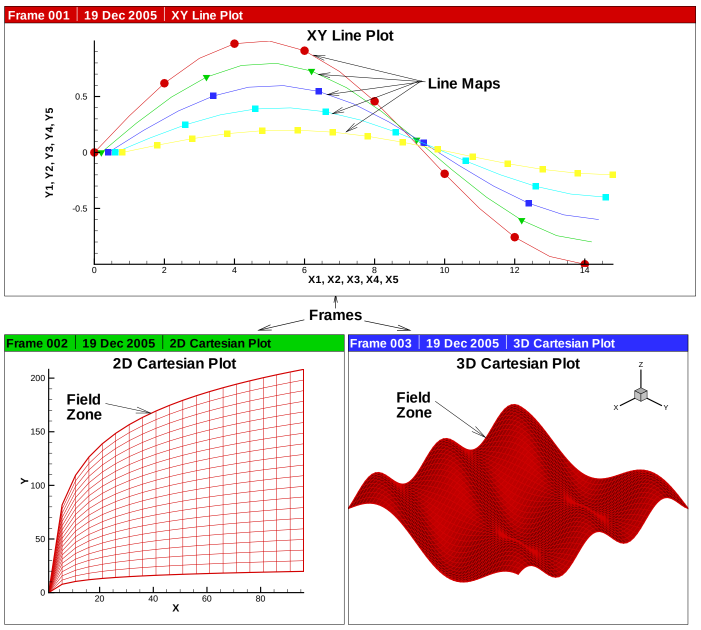

Figure 1: **GAUSSplot** Terms

In line plots (XY Line, Bar, and Polar Line), the graph consists of a collection of one or more **line maps**. In field plots (2D and 3D Cartesian, 2D and 3D Contour), the graph consists of a collection of one or more **field zones**.

To create a graph in **GAUSSplot**, you must first write the data that you want to plot to a **GAUSSplot** format (`.plt`) data file. A **GAUSSplot** format data file consists of one or more **zones** of data, each containing the same number of variables and using the same variable names. In line plots, you may specify **line maps** to use particular variables from any of the data **zones** to associate with the X and Y axes. Field plots contain a **field zone** for each **zone** of data in the data file, and you may specify which variable in the data **zones** is to be associated with each axis in the plot.

Each of the basic plot types has a set of **layers**, which may be used to represent the data. The available layers for each basic plot type are as follows:

	Basic Plot Types			
Layers	XY Line	Polar Line	2D Cartesian	3D Cartesian
Lines	X	X		
Symbols	X	X		
Error bars	X			
Bars	X			
Mesh			X	X
Contour			X	X
Vector			X	X
Scatter			X	X
Shade			X	X
Boundary			X	X

3 The Four Parts of a GAUSSplot Program

There are four steps to creating a graph with **GAUSSplot**:

1. Setup and initialization
2. Creating data and writing the data file
3. Setting various plot options
4. Plotting the graph

These must all be present in any program that creates a graph using **GAUSSplot**, with the exception of the second step if the necessary data file already exists.

To demonstrate each of these steps, let us use the following example of a simple XY Line plot:

```
library gaussplot;
#include gp.sdf

struct gpPlotControl gp;
gp = gpXYLinePlotCreate;

struct gpData gdat;

x = seqa(-pi/2,pi/12,25);
y = sin(x)~cos(x);
string vnames = { "X", "Y1", "Y2" };
```

```
gdat = gpSetPlotData(x~y,vnames);
ret = gpWritePlotData(&gdat,"mydata.plt");

linemaps = { 1 2, 1 3 };
ret = gpSetLineMaps(&gp,1,linemaps);         // Create 2 line maps:
                                             //   X axis   Y axis
                                             //     X        Y1
                                             //     X        Y2
ret = gpSetDataFile(&gp,0,"mydata.plt");
ret = gpShowSymbolLayer(&gp,1,1);            // Show symbols at data points
ret = gpSetSymbolSkip(&gp,1,1|2,2,2);        // Skip every other data point
ret = gpSetSymbolFillMode(&gp,1,1|2,2,"");  // Fill symbols with line color
ret = gpSetSymbolSize(&gp,1,1|2,2);          // Set symbol size to 2% of frame size

ret = gpPlot(&gp);
```

which produces the following graph:

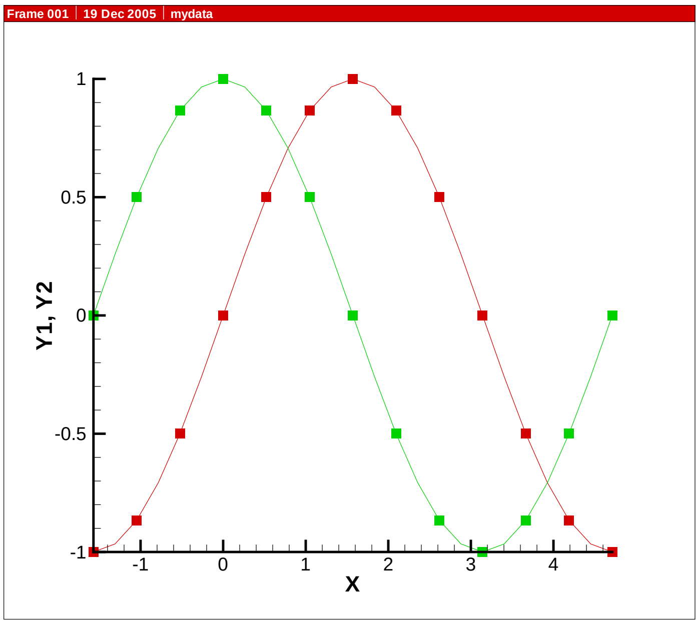

Figure 2: Example XY Line plot graph

3.1 Setup and Initialization

At the beginning of a **GAUSSplot** program, you must activate the **GAUSSplot** library and define and initialize the control structure for your plot:

```
library gaussplot;
#include gp.sdf

struct gpPlotControl gp;
gp = gpXYLinePlotCreate;
```

3.2 Creating Data

If the data file that you want to use in your graph does not already exist, then the next step is to create data. If you are going to use `gpPlot` to plot your graph, you must also write the data file at this point in your program:

```
struct gpData gdat;

x = seqa(-pi/2,pi/12,25);
y = sin(x)~cos(x);
string vnames = { "X", "Y1", "Y2" };

gdat = gpSetPlotData(x~y,vnames);
ret = gpWritePlotData(&gdat,"mydata.plt");
```

3.3 Setting Various Plot Options

GAUSSplot contains many functions that allow you to specify the number of line maps or field zones that will be displayed in your graph as well as how they are to be displayed. These commands are almost all optional; **GAUSSplot** will use defaults for all of the options that you do not set explicitly. The only exception to this is that if you are using `gpPlot` to plot your graph, then you must first call `gpSetDataFile` to specify the data file to be used for the graph.

```
linemaps = { 1 2, 1 3 };
ret = gpSetLineMaps(&gp,1,linemaps);        // Create 2 line maps:
                                            //   X axis   Y axis
                                            //     X        Y1
                                            //     X        Y2
ret = gpSetDataFile(&gp,0,"mydata.plt");
ret = gpShowSymbolLayer(&gp,1,1);           // Show symbols at data points
ret = gpSetSymbolSkip(&gp,1,1|2,2,2);       // Skip every other data point
ret = gpSetSymbolFillMode(&gp,1,1|2,2,"");  // Fill symbols with line color
ret = gpSetSymbolSize(&gp,1,1|2,2);         // Set symbol size to 2% of frame size
```

3.4 Plotting the Graph

Once all of the desired options have been set, you may plot your graph:

```
ret = gpPlot(&gp);
```

To create a simple graph with one frame, containing multiple line maps or a single field zone, you may use an appropriate `gpMake`*PlotType*`Plot` command instead to write the data file, set the line maps, and plot the graph.

The above graph could be created using `gpMakeXYLinePlot` as follows:

```
library gaussplot;
#include gp.sdf
```

```
struct gpPlotControl gp;
gp = gpXYLinePlotCreate;

x = seqa(-pi/2,pi/12,25);
y = sin(x)~cos(x);
string vnames = { "X", "Y1", "Y2" };

ret = gpSetDataFile(&gp,0,"mydata.plt");
ret = gpShowSymbolLayer(&gp,1,1);           // Show symbols at data points
ret = gpSetSymbolSkip(&gp,1,1|2,2,2);       // Skip every other data point
ret = gpSetSymbolFillMode(&gp,1,1|2,2,"");  // Fill symbols with line color
ret = gpSetSymbolSize(&gp,1,1|2,2);         // Set symbol size to 2% of frame size
ret = gpMakeXYLinePlot(&gp,x,y,vnames);
```