数学图表动态生成的 MathGL 应用①

祝晓鹰[1]，彭 燕[1]，余 峰[2]

[1](解放军 63892 部队，洛阳 471003)
[2](解放军 63891 部队，洛阳 471003)

摘 要：结合例程，介绍了数学图形库 MathGL 的若干应用方式。

关键词：MathGL；数学图表；脚本

Application of MathGL in Dynamically Generateing Mathematical Graphics

ZHU Xiao-Ying[1], PENG Yan[1], YU Feng[2]

[1](PLA Unit 63892, Luoyang 471003, China)
[2](PLA Unit 63891, Luoyang 471003, China)

Abstract: MathGL is a library for making mathematical graphics. This paper demonstrates the usage of MathGL with three examples.

Key words: MathGL; mathematical graphics; script

在程序设计中实现各种数学图表的动态生成，常用的有以下几种方法：

① 自己编码

② 调用 Matlab 组件

③ 调用 NI Measurement Studio 控件

自己编码只适合简单的图表，对于复杂应用，例如 3D，由于涉及坐标计算、颜色渲染等，其工作量甚至有可能超过主程序。

Matlab 和 NI 控件都是成熟的商业软件，提供了丰富的数据分析、数据显示功能，可以很方便的集成到用户程序中，缺点是部署时需要安装一个庞大的运行库。

本文要介绍的 MathGL 可以弥补上述方法的不足，它是一个开源的类库，具有如下优点：

① 功能强大，可快速生成各种高质量的 1D、2D、3D 数学图表

② 提供 C/C++、Python、MGL 脚本等多种编程接口，易于使用

③ 部署方便，只需拷贝相关的几个动态库即可。

1 MathGL例图

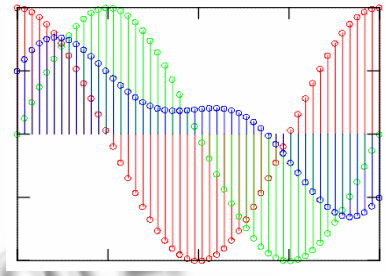

图 1 1D 例图

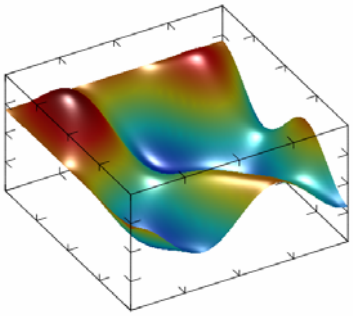

图 2 2D 例图

① 收稿时间:2011-02-15;收到修改稿时间:2011-04-04

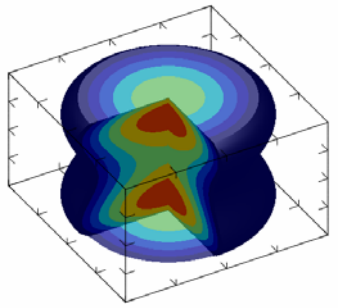

图 3　3D 例图

2　应用MathGL

以下介绍如何在 Visual C++程序中使用 MathGL。

2.1 安装和配置

MathGL 的下载地址：http://mathgl.sourceforge.net/download.html，提供源代码和编译好的版本。由于 MathGL 依赖于另外 5 个开源类库，编译非常繁琐，要求程序员能熟练使用 make 工具，因此建议使用编译好的二进制版本，这里选择 Win32 版。

解压缩下载回来的文件，得到若干子目录，其中 bin 为动态库，include 为 C/C++的头文件，lib 为库文件。添加 bin 目录至系统的 PATH 环境变量。

Win32 版的 MathGL 只提供 MinGW 的库文件（.a），还需要生成 Visual C++可用的库文件（.lib），步骤如下：

① 下载 MinGW 的工具软件 pexports，地址：http://sourceforge.net/projects/mingw/files/MinGW/pexports/

② 运行 pexports，从.dll 文件生成.def 文件：pexports libmgl.dll > libmgl.def

③ 运行 Visual C++的 LIB 工具，从.def 文件生成.lib 文件：

lib /machine:i386/def:libmgl.def

使用上述方法生成的.lib 文件只有标准 C 接口，如果想使用 C++接口，只有重新编译整个类库，这里不再赘述。

前面提到 MathGL 还依赖 5 个类库，它们分别是：

表 1

类库	下载地址	依赖的动态库
giflib	http://gnuwin32.sourceforge.net/packages/giflib.htm	giflib4.dll
gsl	http://gnuwin32.sourceforge.net/packages/gsl.htm	libgsl.dll libgslcblas.dll
libjpeg	http://gnuwin32.sourceforge.net/packages/jpeg.htm	jpeg62.dll
libpng	http://gnuwin32.sourceforge.net/packages/libpng.htm	libpng12.dll
zlib	http://gnuwin32.sourceforge.net/packages/zlib.htm	zlib1.dll

下载上述类库的二进制文件，将依赖的动态库拷贝至 MathGL 的 bin 目录，然后下载 gsl 类库的开发文件，解压缩其中的 include 目录（编译程序时要用）。至此 MathGL 安装完毕。

生成 Visual C++项目时，将 MathGL 和 gsl 的 include 目录添加至项目属性的附加包含目录，将 MathGL 的 lib 目录添加至附加库目录，即可完成编译环境的配置。

2.2 显示 MathGL 图像

MathGL 使用两种方式输出图像：

① 生成图片文件

② 在内存中生成图像的像素数据

方式 1 支持的图片格式见下表：

表 2

图片格式	输出函数
PNG	mgl_write_png
JPEG	mgl_write_jpg
GIF	mgl_write_gif
BMP	mgl_write_bmp
EPS	mgl_write_eps

方式 2 中，像素数据保存在一个二维的字节数组，格式如下：

[0,0]R [0,0]G [0,0]B [1,0]R [1,0]G [1,0]B … …

[0,1]R [0,1]G [0,1]B [1,1]R [1,1]G [1,1]B … …

… …

其中[i, j]R/G/B 为像素[i, j]的红/绿/蓝分量。

本文例程使用方式 2 输出图像，为此笔者编写了一个辅助类 CBitmapForMathGL，用于将像素数据转换成 BITMAP 对象并绘制到 Win32 窗口。

2.3 例程一使用 C 接口

以下代码片断示范如何绘制一条正弦曲线。

包含相关的头文件和库文件：

```
#include "BitmapForMathGL.h"
#include <mgl/mgl_define.h>
#include <mgl/mgl_c.h>
#pragma comment(lib, "libmgl.lib")
```

定义 Graph 对象，它是 MathGL 的内核，实现所有的绘图功能。

```
HMGL gr;
```

生成 Graph 对象，该代码通常位于程序的初始化部分，例如类构造函数：

```
// 初始绘图区域尺寸：300x200
gr = mgl_create_graph_zb(300, 200);
```

绘制图像，该代码通常位于窗口的 OnPaint（WM_PAINT）事件中：

```
void CChildView::OnPaint()
{
CPaintDC dc(this);
// 调整绘图区域
 CRect rt;
 GetClientRect(&rt);
 mgl_set_size( gr, rt.Width(), rt.Height() );
 // 计算正弦曲线的坐标数据
 const int count = 100;
 const double pi = 3.1415926;
 double x[count], y[count];
 for (int i = 0; i < count; i++) {
x[i] = i*1.0/(count-1);
y[i] = sin(2*pi*x[i]);
 }
 // 生成 MathGL 数组对象
 HMDT xdat = mgl_create_data();
 HMDT ydat = mgl_create_data();
mgl_data_set_double(xdat, x, count, 1, 1);
mgl_data_set_double(ydat, y, count, 1, 1);
 // 根据数据调整坐标轴范围
 mgl_set_xrange(gr, xdat, 0);
 mgl_set_yrange(gr, ydat, 0);
 mgl_adjust_ticks(gr, "xy");
 // 绘图
 mgl_axis(gr, "xy"); // 坐标轴
 mgl_box_str(gr, "", true); // 边框
 mgl_plot_xy(gr,xdat,ydat,""); // 正弦曲线
 // 生成 BITMAP 对象
 CBitmapForMathGL mglBitmap(
mgl_get_rgb(gr), mgl_get_width(gr),
mgl_get_height(gr) );
 // 输出到窗口
 mglBitmap.draw(dc.GetSafeHdc());
 // 删除 MathGL 数组对象
 mgl_delete_data(xdat);
 mgl_delete_data(ydat);
}
```

删除 Graph 对象，该代码通常位于程序结束时，例如类析构函数：

```
mgl_delete_graph(gr);
```

例程一运行效果见下图：

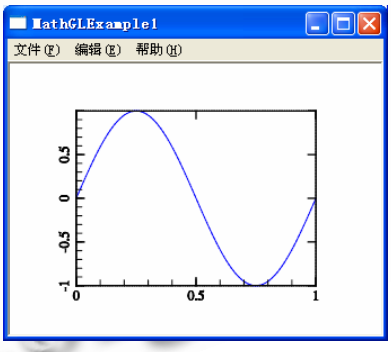

图 4

2.4 例程二使用 MGL 脚本

MathGL 提供了一个简单的脚本语言 MGL，可以完成类库的大部分功能。以下是绘制一组曲线的脚本程序（3curves.mgl）：

```
#生成数据
var x 100 0 1
new y 100 3
put y sin(2*pi*x) -1 0 #曲线 1—sin(2*pi*x)
put y cos(2*pi*x) -1 1 #曲线 2—cos(2*pi*x)
put y x*x*x-12#曲线 3—x*x*x
#调整坐标轴
```

```
xrange x
yrange y
adjust
#绘制
axis#坐标轴
box#边框
plotxy#曲线
```

Visual C++程序调用 MGL 脚本的代码如下：

```
// 调整绘图区域
… …
// 读取脚本文件
std::ifstream in("3curves.mgl");
char mglText[10240];
in.get(mglText, 10240, '\x0');
//创建 Parser 对象
HMPR parser = mgl_create_parser();
//执行脚本
mgl_parse_text(gr, parser, mglText);
//删除 Parser 对象
mgl_delete_parser(parser);
//生成 BITMAP 对象，输出到窗口
… …
```

初始化、释放资源等公共代码参见例程一

例程二运行效果见下图：

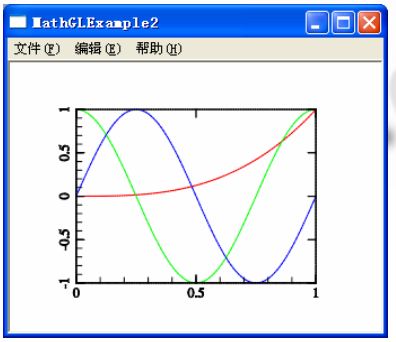

图 5

与 C/C++接口相比，MGL 脚本具有如下优点：

① 代码简洁优雅

② 表达式支持向量计算

③ 绘图模块与主程序松耦合

MathGL 提供了几个命令行工具用于调试运行 MGL 脚本，不过笔者推荐使用 UDAV，一个基于 MathGL 开发的数学工具软件，界面见下图：

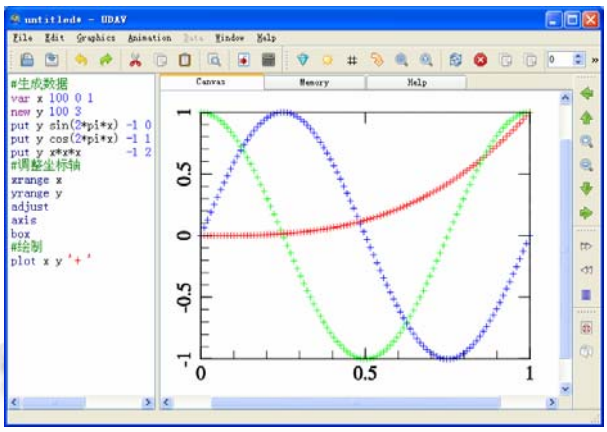

图 6

UDAV 的下载地址：

http://udav.sourceforge.net/download.html

2.5 例程三生成动画

MGL 脚本和主程序之间通过下列两种方式传递数据：

① 主程序调用 API 函数 mgl_add_param 设置参数，脚本程序使用$0、$1 … $9 来引用参数，参数可以是字符串、（脚本中的）变量名或者数值

② 主程序调用 API 函数 mgl_add_var、mgl_data_set_double 创建 MathGL 数组对象并赋初值，脚本程序按名称引用数组

以下代码示范使用 MGL 脚本实现动画，其中用到了上述两种数据传递方式。

脚本程序（animation.mgl）：

```
title '$0'
box
plot dat*sin($1) #数组 dat 来自主程序
```

Visual C++程序：

```
void CChildView::OnTimer(UINT nIDEvent)
{
 //调整绘图区域
 … …
 static double ticker = 0;
 ticker += 0.1;
 //读取脚本文件
```

```
 std::ifstream in("animation.mgl");
 char mglText[10240];
 in.get(mglText, 10240, '\x0');
 //创建 Parser 对象
 HMPR parser = mgl_create_parser();
 //设置脚本参数—$0
 mgl_add_param(parser, 0,
"Simple Animation Demo");
 //设置脚本参数—$1
 char buf[64];
 mgl_add_param( parser, 1,
_gcvt(ticker,9,buf) );
 //设置脚本中的数组变量—dat
 const int count = 100;
 const double pi = 3.1415926;
 double y[count];
 for (int i = 0; i < count; i++) {
double x = i*1.0/(count);
y[i] = sin(2*pi*x + ticker);
 }
 HMDT var = mgl_add_var(parser, "dat");
 mgl_data_set_double(var, y, count, 1, 1);
 //执行脚本
 mgl_parse_text(gr, parser, mglText);
 //删除 Parser 对象
 mgl_delete_parser(parser);
 //生成 BITMAP 对象
 CBitmapForMathGL mglBitmap(
mgl_get_rgb(gr), mgl_get_width(gr),
 mgl_get_height(gr) );
 //输出到窗口
  CClientDC dc(this);
 mglBitmap.draw(dc.GetSafeHdc());
 }
```

初始化、释放资源等公共代码参见例程一，Windows 定时器的使用请参考相关书籍。

例程三运行效果见下图：

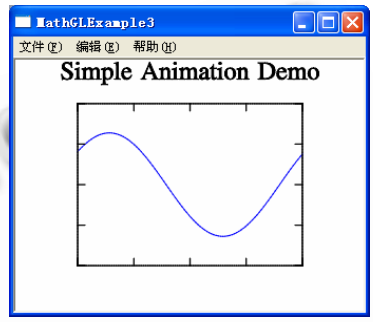

图 7

3 结语

以上介绍了 MathGL 几种常见的应用方式，其中 MGL 脚本由于实现了绘图模块与界面的松耦合，推荐开发人员首选。

为方便读者，本文例程已放在互联网上，下载地址：http://ishare.iask.sina.com.cn/f/13201233.html

参考文献

1 Balakin A. MathGL Manual Book. 2008.

2 潘爱民,王国印译.VisualC++技术内幕.第 4 版.北京:清华大学出版社,1999.