意圖[鴻蒙版]

(SVGLABv1.0)

使用手册

作者：張士超

zhshchao@163.com

第一章 前言

1.1、寫此模塊的原因

perl 原有的 SVG 模塊使用複雜，且做許多可由程序完成的工作。

1.2、名稱的意義

1，SVG 與 MATLAB 的合義，以 MATLAB 風格語句來畫 SVG 圖像；
2，兼取 SVG 實驗室之意；
3，沒有接觸過 SVG，沒有使用過 MATLAB，不影響使用此模塊。

1.3、指導思想

下最少的定義，做最多的事情。

1.4、定義與函數名

SVG 的如直線、圓等一個獨立的組件稱為元素。
除個別，一切作圖或屬性設置函數均以 SVG 元素名，或 MATLAB 相應函數名為名。

1.5、安裝

將模塊拷入 perl 可訪問的模塊目録即可。
事實上，如下代碼給出所在機器上所有 perl 可訪問的模塊目録：
perl -e 'print"@INC"';

如有程序運行異常，手册所釋有乖、行文不暢，乃至錯別字等，一請致信作者。

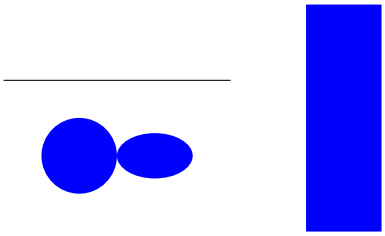
第二章 快速使用

在作圖前需要知道的是，模塊定義 Y 軸向上；默認時一個單位為一個像素。

2.1、作一張圖

一個 figure-figend 對生成一張圖片：

```
#!/usr/bin/perl

use SVGLAB;

figure;
        line(100,300,400,300);      #直線
        rect(500,400,100,300);      #矩形
        circle(200,200,50);         #圓
        ellipse(300,200,50,30);     #橢圓
figend;
#生成 SVGLAB1.svg
```

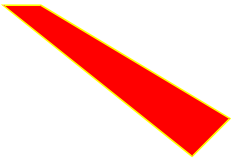


第一個實例

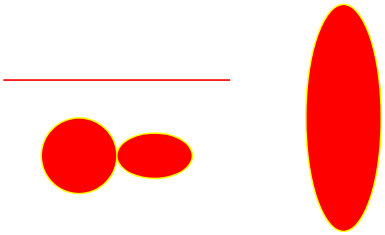
2.2、做多張圖

多個 figure-figend 對生成多張圖片：

```
#!/usr/bin/perl
use SVGLAB;

figure;
        $x=[200,450,500,250];
        $y=[300,100,150,300];
        polygon($x,$y,'red',2,'yellow');#紅內黃邊多邊形
figend;
#將生成 SVGLAB1.svg
figure;
        line(100,300,400,300,2,'red');      #紅色直線
        rect(500,400,100,300,10,10,'red',2,'yellow');   #紅內黃邊彎角矩形
        circle(200,200,50,'red',2,'yellow');        #紅內黃邊圓
        ellipse(300,200,50,30,'red',2,'yellow');   #紅內黃邊橢圓
figend;
#將生成 SVGLAB2.svg
```



多邊形



一張圖上的多種元素

2.3、設置圖片參數

2.3.1、局部設置－figure()函數

1，figure()設置圖片大小與名稱

```
#!/usr/bin/perl
use SVGLAB;

figure(500,500);#設置圖片大小為 500*500 像素
        rotate(45,300,200);#以點(300,200)為心正旋 45 度
        shq("fill-opacity='0.5'");#透明度為 0.5
        rect(300,400,100,200);
figend;
#SVGLAB1.svg,500*500
```

```
#!/usr/bin/perl
use SVGLAB;

figure('mySVG.svg');#或 figure('/home/usr/SVGLAB_eg/mySVG.svg')
        rotate(45,300,200);#以點(300,200)為心正旋 45 度
        shq("fill-opacity='0.5'");#透明度為 0.5
        rect(300,400,100,200);
figend;
#mySVG.svg,800*500
```

SVGLAB1.svg,500*500

mySVG.svg,800*500

也可以 figure('名稱',寬,高)同時設置大小與名稱。

2，figure()使用方法詳介：

```
figure;
figure(寬,高);
figure('名稱');
figure('名稱',寬,高)
```

2.3.2、全局設置－SVGLAB()函數

1，SVGLAB()設置圖片大小與前綴：

SVGLAB()可以設置全局大小與前綴，如：

```
#!/usr/bin/perl
use SVGLAB;
SVGLAB('mySVG');
        figure;
        figend;

        figure;
        figend;
SVGLAB('smallSVG',200,100);
        figure;
        figend;

        figure;
        figend;
```

將產生四個空白文件：
mySVG1.svg、mySVG2.svg，800*500 像素;
smallSVG1.svg、smallSVG2.svg，200*100 像素.

2，SVGLAB()使用方法詳介：
SVGLAB()用於 figure-figend 對之外，4 種調用方法：

```
SVGLAB;                 #使狀態回復默認:寬 800，高 500，前綴'SVGLAB';
SVGLAB(寬,高);          #如 SVGLAB(2000,1500);寬 2000，高 1500，前綴'SVGLAB';
SVGLAB('前綴');         #如 SVGLAB('mySVG');寬 800，高 500，前綴'mySVG';
SVGLAB('前綴',寬,高);          #如 SVGLAB('mySVG',2000,1500);順序不可改易.
```

第三章 SVG 元素與屬性

3.1 元素

確切來說，3.11 至 3.14 並非 SVG 元素，但它們在使用上與 SVG 元素風格一致，且為述說方便，一並列入此節。

3.1.1、直線

line($x1,$y1,$x2,$y2,線寬,線色);#($x1,$y1),($y1,$y2)分別為直線起止點
最簡調用：
line($x1,$y1,$x2,$y2);
（黑色，線寬為 1）

3.1.2、矩形

rect($x,$y,寬,高,x 彎角,y 彎角,色,線寬,線色);#($x,$y)是其左上點坐標
最簡調用：
rect($x,$y,寬,高);
（藍色，線寬 0，無線色，彎角 0）

3.1.3、圓

circle($cx,$cy,$r,色,線寬,線色);#($cx,$cy)為圖心，$r 為半徑
最簡調用：
circle($cx,$cy,$r);
（藍色，線寬 0，無線色）

3.1.4、橢圓

ellipse($cx,$cy,$rx,$ry,色,線寬,線色);#($cx,$cy)為橢圖心，$rx,$ry 為其 x,y 嚮半徑
最簡調用：
ellipse($cx,$cy,$rx,$ry);
（藍色，線寬 0，無線色）

3.1.5、折線

polyline(x 諸坐標,y 諸坐標,線寬,線色,樣式);
"x 諸坐標"是指一維數組的引用，存儲拆線諸點的 x 坐標，"y 諸坐標"類似
如：
$x=[200,450,500,250];
$y=[300,100,150,300];
polyline($x,$y,'red',2,'yellow');

3.1.6、多邊形

polygon(x 諸坐標,y 諸坐標,線寬,線色,樣式);
用法與 polyline 幾同。

3.1.7、路徑（暫闕）

此版本尚未加入路徑元素，因此用此模塊畫不了扇形（shq 情況除外）。

3.1.8、文本

text(文字,$x,$y,大小,字體,字色,線色);
線色謂文字外圍色
最簡調用：
text(文字,$x,$y)
（14 號，Arial，黑，黑）

3.1.9、鏈接

添加圖片超鏈接，使用方法：
alink('地址','方式');
　一些 SVG 元素
aend;
使 alink-aend 對之間的元素鏈接到'地址'，並以'方式'之方式打開。SVG 提供四種'方式'，'new'、'replace'、'_black'、'_top'，對不同的查看器産生的效果不同，因此建議不設。
最簡調用：
alink('地址');
aend;
注：alink 産生的是 SVG <a>元素，因為'a'字作函數名太短，重名率高，故用'alink'命名。

3.1.10、組

創建一個組元素，使用方法：
gp;
　一些 SVG 元素
gpend;
在 gp 與 gpend 之間的元素將屬於同一個組，可以給這個組的元素設置同樣的屬性，如顏色、旋轉角度。

3.1.11、豎文本

vtext(文字,$x,$y,大小,字體,字色,線色);
同（3.1.8、文本），豎排

3.1.12、plot 函數圖像

plot(x 諸坐標,y 諸坐標,線寬,線色,樣式);
如有變量：

```
$x=linspace(0,6.28,100);
$y=vsin($x);#以上兩函數 SVGLAB 內置
```
典型調用 1：
```
plot($x,$y);
```
（線寬 1，藍）
典型調用 2：
```
plot($x,$y,'','','*');
```
（以*號畫諸點，'樣式'可以為任意字符，或串）

3.1.13 stem 函數圖像

stem(x 諸坐標,y 諸坐標,線寬,線色);
用法與 plot 幾同。

3.1.14 標籤

```
xlabel(文字,大小,字體,$x,$y,字色,線色);
ylabel(文字,大小,字體,$x,$y,字色,線色);
title(文字,大小,字體,$x,$y,字色,線色);
```
最簡調用：
```
xlabel(文字);      #默認在圖片下方正中
ylabel(文字);      #默認在圖片左方正中
title(文字);       #默認在圖片上方正中
```

之所以在有了 text()元素之後還要寫標籤，是因為在某些場合（如用 axis(x1,x2,y1,y2)手動設置了坐標範圍之後）會將圖像上下左右邊界覆蓋，這樣無論甚麼元素包括 text()在內都被蓋住了，這時就用標籤來在邊界區域寫文字。

3.2 屬性

3.2.1 元素屬性

1，rotate(角度,x,y);
作用：使其下方首個執行的 SVGLAB 元素繞(x,y)正轉（順時針）相應角度。例子可以參見第二章，2.3.1 節。
2，shq('屬性');
作用：使'屬性'應用於其下方首個執行的 SVG 元素。詳細使用見第五章。

注：SVG 中鏈接雖近乎'屬性'，但它是以元素形式出現的，故不列於此。

3.2.2 圖片屬性

3.2.2.1 局部圖片屬性

1，figure()設置圖片大小與名稱
使用方法四種詳介：
```
figure;                  #默認情況，800*500 像素，SVGLAB1.svg
figure(寬,高); #設置文件大小
figure('名稱');#設置文件名
figure(寬,高,'名稱');
```
2，axis()設置圖片坐標
本模塊的 Y 軸向上，不同於 SVG 固有之向下，事實上程序是將用戶以 Y 軸向上的輸入數據進行了坐標轉換再寫進 SVG 圖片的。若必須使用 SVG 固有坐標時，參見下一節中大 AXIS。
使用方法四種詳介：
```
axis;                    #自適應坐標，使坐標適應此語句上方的所有 SVG 元素。
axis(1);                 #保持比例縮放，使坐標適應此語句上方的所有 SVG 元素。
axis(x1,x2,y1,y2);       #設置坐標範圍.此項會將圖片上下左右邊界覆蓋，以防止圖像畫出縮
```

#放（自動）範圍外。若不希望被覆蓋，可在大 AXIS（見下節）中設置
axis('on'); #顯示坐標軸

若一個 figure-figend 對中有多個 axis 更改坐標的命令（即除以上四種中之'axis('on')'外之三種），則在最後者生效。建議一個 figure-figend 對中衹用一次該種命令。

注：關於 axis()的詳細使用及效果，可以參見 test.pl 附於本文檔後者。

3，數學作圖的 **grid** 屬性

使用方法五種詳介：

```
grid;          #或 grid('')或 grid('xy');
grid('x');     #垂直於 x 軸的網格，對齊到標尺
grid('y');     #垂直於 y 軸的網格，對齊到標尺
grid(m,n);     #分別垂直於 x,y 軸均勻 m,n 條網格，未必對齊到標尺
grid(m);       #分別垂直於 x,y 軸均勻 m,m 條網格，未必對齊到標尺
```

例子可參見第四章，4.3.1。

3.2.2.2 全局圖片屬性

所謂全局圖片屬性，是說定義於 figure-figend 對外，設置後對其下所有圖片均生效，直到新的屬性被設置。

1，**SVGLAB** 設置圖片大小與前綴

使用方法四種詳介：

```
SVGLAB;              #使狀態回復默認:寬 800，高 500，前綴'SVGLAB';
SVGLAB(寬,高);       #如 SVGLAB(2000,1500);寬 2000，高 1500，前綴'SVGLAB';
SVGLAB('前綴');      #如 SVGLAB('mySVG');寬 800，高 500，前綴'mySVG';
SVGLAB('前綴',寬,高);        #如 SVGLAB('mySVG',2000,1500);順序不可改易.
```

2，**AXIS** 設置全局坐標屬性

調用方法 5 種:

```
AXIS;          #默認情況，或回復默認
AXIS(0);       #用 SVG 固有坐標系,不進行任何坐標轉換操作
AXIS(num);  #如 AXIS(0.8);置區域縮放系數為 0.8
AXIS(bool,bool,bool,bool)  #意為 AXIS(是否轉換坐標系,縮放坐標時是否保持縱橫比,是否縮放區域,縮放區域後是否覆蓋邊界)
AXIS(num,bool,bool,bool)#意為 AXIS(區域縮放系數[故轉換坐標為真,縮放區域為真],縮放坐標時是否保持縱橫比,是否縮放區域[此項無效,定真],縮放區域後是否覆蓋邊界)如:
AXIS(0.5,1,1,1);       #AXIS(0.5,1,0,1)將產生相同效果
```

注：關於 SVGLAB()與 AXIS()的詳細使用及效果，可以參見 test.pl 附於後者。

第四章 數學作圖

MATLAB 有方便與強大的數學作圖功能，這裏簡單模仿了一些。

4.1、向量生成

在使用數學函數前需要說明的是，向量的定義。由於 MATLAB 向量的定義形式恰好與 perl 一維數組的引用一致，故這裏以此作為向量的定義，如：

```
$x=[1,2,3,4,5];
```

定義一個向量。（衹定義了行向量）

目前提供 vector()和與 MATLAB 相同的 linspace()函數：

```
$x=vector(起,步長,止);          #相當於 MATLAB 中 x=起:步長:止
$x=linspace(起,止,長度);        #與 MATLAB 用法同
```

另外，$x=[a..b];可以生成步長為 1 的整數向量。

4.2、數學函數

目前提供如下數學函數：

```
$y=vsin($x)              #生成向量$y，每一維為$x 相應維的正弦值
$y=vcos($x)              #生成向量$y，每一維為$x 相應維的餘弦值
$y=vpoisson($lamda,$k)#$k 的諸維為整數，生成向量$y，每一維為$k 相應維的泊松分布概率
```

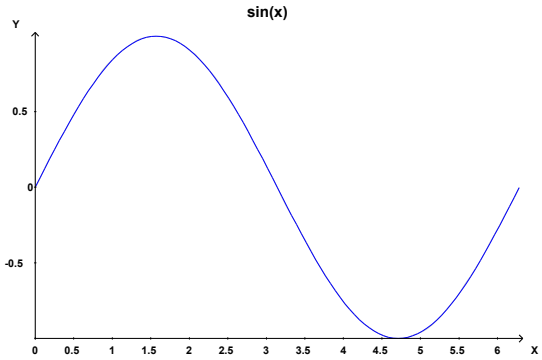
函數名冠以 v 者，為避 perl 自帶函數名，且明是向量（vector）意。

4.3、作圖與屬性

目前提供兩個類似於 MATLAB 的作圖函數，plot 與 stem 的使用參數詳見第三章，3.1.下面是兩個例子：

4.3.1、plot 作圖例

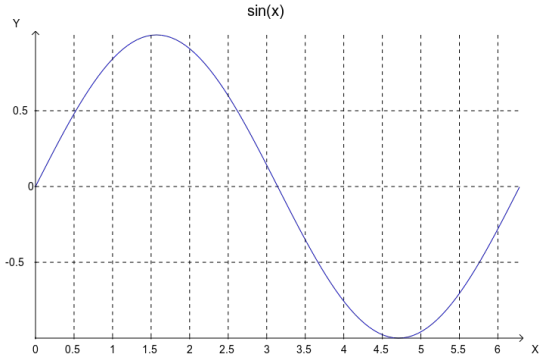
```
#!/usr/bin/perl
use SVGLAB;

$x=linspace(0,6.28,100);#從 0 到 6.28,100 個點
$y=vsin($x);
figure;
        plot($x,$y);
        axis;
        axis('on');
        title('sin(x)');
figend;
```



plot 作正弦圖像

除了 axis 與 title 外，還可以使用 grid 屬性：

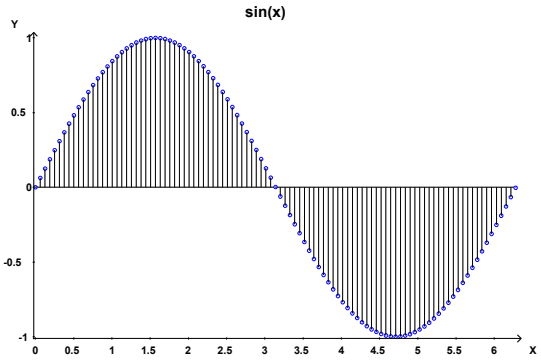
```
#!/usr/bin/perl
use SVGLAB;

$x=linspace(0,6.28,100);#從 0 到 6.28,100 個點
$y=vsin($x);
figure;
        plot($x,$y);
        axis;
        axis('on');
        title('sin(x)');

        grid;
figend;
```



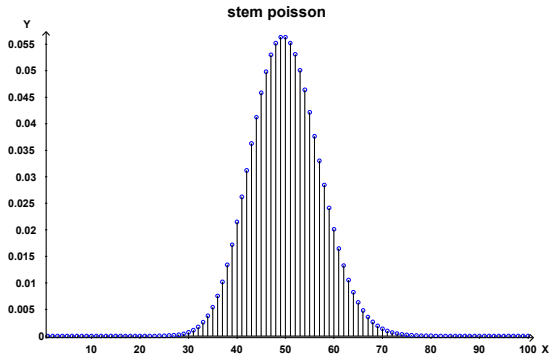
帶網格的正弦圖像

4.3.2、 stem 作圖例

```
#!/usr/bin/perl
use SVGLAB;
$x=linspace(0,6.28,100);#從 0 到 6.28,100 個點
$y=vsin($x);
figure;
        stem($x,$y);
        axis;
        axis('on');
        title('sin(x)');
figend;

$x=[1..100];
$y=vpoisson(50,$x);
figure;
        stem($x,$y);
        axis;
        axis('on');
        title('stem poisson');

figend;
```



stem 畫正弦圖像



stem 畫泊松分布

第五章 關於特效－－神祇函數

SVG 之所以流行，原因之一是其豐富的圖片特效。

本模塊指導思想是下盡量少的定義，這與 SVG 的豐富特效無疑是一個衝突。

為了解決此問題，在盡量少增加定義，且不增加函數參數（否則會使參數繁鎖難以記憶）的情況下同樣能使用特效，設計了神祇－shq 函數。

5.1、一些 shq 效果

5.1.1、shq 設置元素屬性

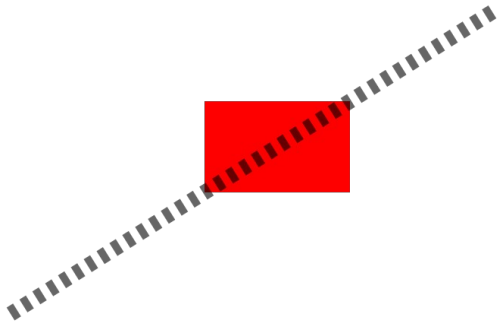
shq 將屬性賦予其下距其最近的 SVGLAB 元素，屬性采用 SVG 原始的定義方式。如下面兩例：

1，虛線與線透明

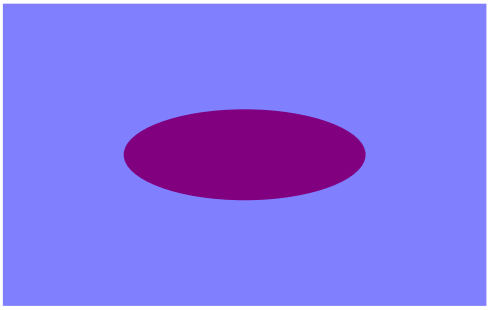
```
#!/usr/bin/perl
use SVGLAB;
figure;
        rect(1.4,1.7,0.3,0.3,'','','red');      #參考矩形

        shq('stroke-dasharray="10"');     #虛線
        shq('opacity="0.6"');                   #線透明
        line(1,1,2,2,20);       #線寬為 20

        axis;                   #自適應坐標（未必保持橫縱比）
figend;
```

2，填充透明

```
#!/usr/bin/perl
use SVGLAB;

figure;
        ellipse(2,2,0.5,0.3,'red');    #參考橢圓
        shq("fill-opacity='0.5'");
        rect(1,3,2,2);
        axis;#自適應坐標
figend;
```



shq 實現虛線和線透明



shq 實現矩形透明

我們看到，在不增加函數參數的情況下，shq 可以把各種屬性應用到元素上，正因其有這樣的特質，故得神祇之名。

5.1.2、shq 特效

事實上，shq 並未使特效的生成有所簡化，這裏祇是想說明用 shq 可以實現這些特效，希望以後能找到更為方便的方法。

5.1.2.1 漸變

把用 shq 定義好的"id"項加#號放到'url()'內,作為顏色傳入圖形。這也是 SVG 自身的做法。

1，線形漸變

```
#!/usr/bin/perl
use SVGLAB;
figure;
        shq('<defs>
                <linearGradient id="orange_red" x1="0%" y1="0%"
                        x2="100%" y2="0%">
                <stop offset="0%" style="stop-color:rgb(255,255,0);
                stop-opacity:1"/>
                <stop offset="100%" style="stop-color:rgb(255,0,0);
                stop-opacity:1"/>
                </linearGradient>
                </defs>
        ');#以上漸變定義拷自:
          #http://www.w3school.com.cn/svg/svg_grad_linear.asp

        rect(1,3,2,2,0.2,0.2,'url(#orange_red)');
        axis;#自適應坐標
figend;
```

2，放射漸變

```
#!/usr/bin/perl
        use SVGLAB;
        figure;
                shq('<defs>
                        <radialGradient id="grey_blue" cx="50%"
                                cy="50%" r="50%"
                        fx="50%" fy="50%">
                        <stop offset="0%" style="stop-color:rgb(200,200,200);
                        stop-opacity:0"/>
                        <stop offset="100%" style="stop-color:rgb(0,0,255);
                        stop-opacity:1"/>
                        </radialGradient>
                        </defs>
                ');#以上漸變定義拷自:
                  #http://www.w3school.com.cn/svg/svg_grad_radial.asp

                rect(1,3,2,2,0.2,0.2,'url(#grey_blue)');
                axis(1);#自適應坐標，保持橫縱比
        figend;
```

線形漸變色

放射漸變色

5.1.2.2 濾鏡

把用 shq 定義好的濾鏡"id"項加#號放到'url()'內，放入之後的 shq()裏，傳入圖形元素。因為濾鏡

不是一種顏色，所以與漸變的處理方法有異。

```
#!/usr/bin/perl
use SVGLAB;
figure;#放射漸變      1
        #高斯濾鏡 3
        shq('<defs>
                <filter id="Gaussian_Blur">
                <feGaussianBlur in="SourceGraphic" stdDeviation="10"/>
                </filter>
                </defs>
        ');#以上濾鏡定義拷自:
          #http://www.w3school.com.cn/svg/svg_filters_gaussian.asp

        shq('filter="url(#Gaussian_Blur)"');
        rect(1,3,2,2,0.2,0.2);
        axis;#自適應坐標
figend;
```

高斯濾鏡

以下是 SVG 定義的一些濾鏡:

* feBlend
* feColorMatrix
* feComponentTransfer
* feComposite
* feConvolveMatrix
* feDiffuseLighting
* feDisplacementMap
* feFlood
* feGaussianBlur
* feImage
* feMerge
* feMorphology
* feOffset
* feSpecularLighting
* feTile
* feTurbulence
* feDistantLight
* fePointLight
* feSpotLight

5.1.2.3 動畫

```
#!/usr/bin/perl
use SVGLAB;
```

```
AXIS(0);#還原 SVG 固有坐標
figure(240,240);#SVG 時鐘
        shq('transform="translate(120,120)  rotate(180)"');
        gp;

                circle(0,0,102,'none',2,'green');#鐘框

                shq('   <animateTransform attributeName="transform"
                         type="rotate"
                        repeatCount="indefinite" dur="12h" by="360" />
                ');
                gp;#時針
                        shq('opacity="0.5"');
                        line(0,0,0,80,5,'black');
                        circle(0,0,7,'black');
                gpend;
                shq('   <animateTransform attributeName="transform"
                         type="rotate"
                        repeatCount="indefinite" dur="60min" by="360" />
                ');
                gp;#分針
                        shq('opacity="0.9"');
                        line(0,0,0,95,4,'red');
                        circle(0,0,6,'red');
                gpend;
                shq('   <animateTransform attributeName="transform"
                         type="rotate"
                        repeatCount="indefinite" dur="60s" by="360" />
                ');
                gp;#秒針
                        line(0,0,0,100,2,'blue');
                        circle(0,0,4,'blue');
                gpend;
        gpend;
        axis;

figend;         #以上時鐘模仿自：http://kb.operachina.com/node/161
```

SVG 動畫推薦用 opera9 以上版本查看。

5.1.3、shq 生成元素

可以將元素的 SVG 描述由 shq 直接寫進圖片。甚至，可以用 shq 來生成整張圖片：

```
#!/usr/bin/perl
use SVGLAB;
figure;#shq 生成整張圖片
        shq('<line x1="100" y1="200" x2="400" y2="200" stroke-width="1"
stroke="black" ></line>');
        shq('<rect x="500" y="100" width="100" height="300" rx="0" ry="0"
fill="blue"  ></rect>');
        shq('<circle cx="200" cy="300" r="50"  fill="blue"  ></circle>');
        shq('<ellipse cx="300" cy="300" rx="50" ry="30"  fill="blue"
></ellipse>');
figend;
```

5.2、shq 原理

shq 的原理是將字符串插入到合適的位置：若 shq 的參數是元素屬性，則將其插入到[元素內]；若是動畫描述，則將其插入到[元素內另一個位置]；若是特效定義，則將其寫到[元素外]。

上述諸位置，如下 SVG 文件所示：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="800" height="500" xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink">
<line x1="80" y1="450" x2="720" y2="250" stroke-width="1" stroke="black" [元素内]>
[元素内另一個位置]</line>
[元素外]
<line x1="80" y1="450" x2="720" y2="50" stroke-width="1" stroke="black" ></line>
        <!--
        Generated using the Perl SVGLAB Module V1.0
        by Zhang Shichao
        email:zhshchao@163.com
 -->
</svg>
```

5.3、 大 ShQ

甚麼是大 ShQ 呢？shq 雖然比較方便，但有時 SVG 的描述很長，難以記住，比如說虛線的定義為'stroke-dasharray'，要輸入也比較繁鎖。因此建立在小 shq 的基礎之上的大 ShQ 應運而生，它簡化了一些繁鎖屬性的名字，如用'xx'（虛線）代表'stroke-dasharray'，如下命令：

ShQ('xx:10')

相當於調用了 shq('stroke-dasharray="10"');

而 ShQ('xx:10;xtm:0.6')

相當於調用了兩個 shq:

shq('stroke-dasharray="10"')與 shq('opacity="0.6"')（xtm，線透明）;

目前，祇規定了三個常用的屬性名：

xx（虛線）　　==stroke-dasharray

tm（透明）　　==fill-opacity

xtm（線透明）　==opacity

ShQ 屬性與屬性值之間用英文":"號隔開，各個屬性之間以英文分號（可以加任意空白，包括換行）隔開。

以後如有需要，會加入更多的簡稱名。

第六章 不足與改進方向

不足與所需的改進有以下幾個方面：
一、支持更多 SVG 圖形和效果
二、對更多 MATLAB 作圖的模仿
三、對 MATLAB 數學支持的模仿，如列向量，矩陣。
四、更多高層函數的開發

7.1 支持更多 SVG 圖形和效果

對 path 的支持
扇形函數的編寫
三角形函數
ShQ 簡稱名的增加

7.2 對更多 MATLAB 作圖的模仿

subplot（目前實際上有一個 subplot 函數，但功能不完備）
句柄的支持
3D 作圖（這個雖然比較困難，依然列在此）

7.3 對 MATLAB 數學支持的模仿

vexp
向量的運算（將不得不定義一些新函數）

7.4 更多高層函數的開發

餅圖
工作中常用的圖，甚至是常見形式文件的數據讀取，畫其圖像。

附錄一 中文版本號說明

中文版本號分兩位。
高位以千字文序：天地玄黃，……，焉哉乎也；
低位以天干序：甲乙丙丁戊己庚辛壬癸；
低位甲增至癸之際，高位易字，低位復甲。

與英文版本號之對應：
低位甲對 1，字易數增；高位天對 1，字易數增。
如：天甲 == v1.1；天乙 == v1.2；天癸 == v2.0；
　　地甲 == v2.1；地丙 == v2.3；宙辛 == v6.8．

v1.0 版實為首版，稱“鴻蒙”版。
天甲又可稱天字一版；
宙辛又可稱宙字八版。

附錄二 幾個例子

例一、**axis** 自適應坐標系

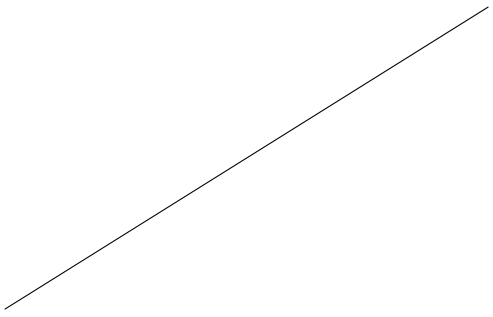
用 SVGLAB 作圖一個方便之處是可以不必關心圖片的像素，甚至可以給元素的坐標賦以負值，如下例將一條直線畫在第四象限：

```
#!/usr/bin/perl
use SVGLAB;
figure;
        line(-1,-1,0,0);    #從(-1,-1)到(0,0)的一條直線
        axis;                #自適應
figend;
```

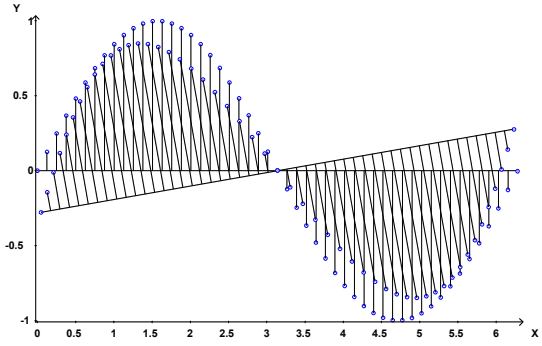
例二、**rotate** 作用於元素

一般而言，屬性可以作用於任何元素，如下例，rotate 作用於 stem 元素：

```
#!/usr/bin/perl
use SVGLAB;
figure;
        $x=linspace(0,6.28,50);
        $y=vsin($x);
        stem($x,$y);
        rotate(-10,3.14,0);   #以(3.14,0)為心逆轉 10 度。
        stem($x,$y);
        axis;                 #自適應
        axis('on');           #畫坐標軸
figend;
```



axis 自適應坐標例



rotate 例

例三、組的使用

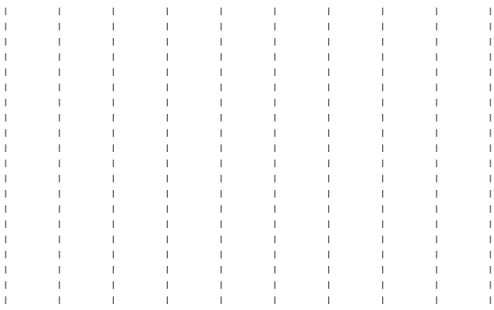
此例為屬性作用於組元素，如可以給 10 條直線加上虛線屬性：

```
#!/usr/bin/perl
use SVGLAB;

    figure;
        ShQ('xx:10'); #或完整的 shq('stroke-dasharray="10"');
        gp;             #將如下 10 條直線合為一組
        for$i(1..10){
            line($i,1,$i,10);
        }
        gpend;          #組結束
        axis;
    figend;
```

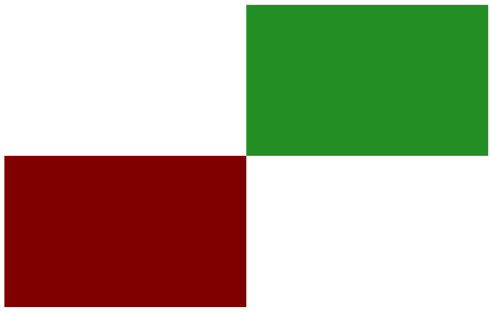
例四、顏色值

顏色值是 SVG 支持的值，如'red'、'rgb(128,0,0)'、'#238e23'等，如：

```
#!/usr/bin/perl
use SVGLAB;
    figure;
        rect(1,3,2,2,'','','rgb(128,0,0)');
        rect(3,5,2,2,'','','#238e23');
        axis;
    figend;
```



組內元素應用相同屬性例



顏色值例

附錄三 幾個重要函數的測試

文件名：test.pl
文件内容：

```
#!/usr/bin/perl -w
use strict;
use utf8;
use SVGLAB;

my ($x,$y);


=c
測試方法：
編寫測試某功能的代碼至一子過程
該子過程的調用語句在該子過程之下
釋放調用語句以測試


測試結果說明：
本文件中列出所有測試項均已測試；
結果達預期.
=cut

##====================axis 測試====================##
=c      axis;                   #自適應坐標，使坐標適應此語句上方的所有 SVG 元素。
axis(1);                        #保持比例縮放，使坐標適應此語句上方的所有 SVG 元素。
axis(x1,x2,y1,y2);    #設置坐標範圍.此項會將圖片上下左右邊界覆蓋，以防止圖像畫出縮放（自動）範圍外。若不希望被覆蓋，可在大 AXIS（見下節）中設置
axis('on');             #顯示坐標軸
若一個 figure-figend 對中有多個 axis 更改坐標的命令（即除以上四種中之'axis('on')'外之三種），則在最後者生效。建議一個 figure-figend 對中衹用一次該種命令。

=cut
sub axis_test{
        figure;#SVGLAB1.svg
                $x=linspace(0,6.28,100);#從 0 到 6.28,100 個點
                $y=vsin($x);
                plot($x,$y);     #坐標太小,看不到
        figend;

        figure;#SVGLAB2.svg
                $x=linspace(0,6.28,100);
                $y=vsin($x);
                plot($x,$y);
                axis;#自適應(寬高分別為圖片的 0.8,若不設區域縮放系數，則 axis()將之設定為 0.8)
        figend;#結果:圖片中央一周期正弦曲線

        figure;#SVGLAB3.svg
                $x=linspace(0,6.28,100);
                $y=vsin($x);
                plot($x,$y);
                axis;#自適應(寬高分別為圖片的 0.8)
                axis('on');#顯示坐標軸.
        figend;#結果:圖片中央一周期正弦曲線,帶有坐標軸.

        figure;#SVGLAB4.svg
                $x=linspace(0,6.28,100);
                $y=vsin($x);
```

```
		plot($x,$y);
		axis(0,3.14,0,1);#自適應(寬高分別為圖片的 0.8)
		axis('on');#顯示坐標軸.
	figend;#結果:圖片中央半周期正弦曲線,帶有坐標軸.

	figure;#SVGLAB4.svg
		$x=linspace(0,6.28,100);
		$y=vsin($x);
		plot($x,$y);
		axis(0,3.14,0,1);	#自適應(寬高分別為圖片的 0.8)
		axis('on');		#顯示坐標軸.
		axis(1);		#保持比例
	figend;#結果:圖片中央半周期正弦曲線,帶有坐標軸.
}
#axis_test();
##====================axis 測試畢====================##



##====================SVGLAB 測試====================##
=c
SVGLAB()用於 figure-figend 對之外，4 種調用方法：

SVGLAB;			#使狀態回復默認:寬 800，高 500，前綴'SVGLAB';
SVGLAB(寬,高);		#如 SVGLAB(2000,1500);寬 2000，高 1500，前綴'SVGLAB';
SVGLAB('前綴');		#如 SVGLAB('scaffold');寬 800，高 500，前綴'scaffold';
SVGLAB(寬,高,'前綴');		#如 SVGLAB('scaffold',2000,1500);順序不可改易.
=cut
sub svglab1{
	SVGLAB;#默認狀態,可不寫
		figure;
		figend;

		figure;
		figend;

		figure;
		figend;
#結果應為：三個文件，名為 SVGLAB1.svg、SVGLAB2.svg、SVGLAB3.svg，大小 800*500 像素.

}
#svglab1();
sub svglab2{
	SVGLAB(2000,1500);
		figure;
		figend;

		figure;
		figend;

	SVGLAB(200,100);
		figure;
		figend;

		figure;
		figend;
#結果應為：兩個文件：SVGLAB1.svg、SVGLAB2.svg，大小為 200*100;
#由於 SVGLAB 會重置文件編號,故前兩張大小為 2000*1500 者被覆蓋;
}
```

```
#svglab2();
sub svglab3{
	SVGLAB('scaffold');
		figure;
		figend;

		figure;
		figend;
	SVGLAB('contig');
		figure;
		figend;

		figure;
		figend;
#結果應為：四個文件：scaffold1.svg、scaffold2.svg
#	contig1.svg、contig2.svg，大小均為 800*500.
}
#svglab3();
sub svglab4{
	SVGLAB('scaffold',2000,1500);
		figure;
		figend;

		figure;
		figend;
	SVGLAB('contig',200,100);
		figure;
		figend;

		figure;
		figend;
#結果應為：四個文件：scaffold1.svg、scaffold2.svg，大小為 2000*1500;
#	contig1.svg、contig2.svg，大小為 200*100.
}
#svglab4();
sub svglab5{
	SVGLAB('scaffold',2000,1500);
		figure;
		figend;

		figure;
		figend;
	SVGLAB;			#回復默認
		figure;
		figend;

		figure;
		figend;
#結果應為：四個文件：scaffold1.svg、scaffold2.svg，大小為 2000*1500;
#	SVGLAB1.svg、SVGLAB2.svg，大小為 800*500.
}
#svglab5();
##==================SVGLAB 測試畢==================##

##==================AXIS 測試==================##
=c
調用方法 5 種:
1	AXIS;		#默認情況，或回復默認
2	AXIS(0);	#用 SVG 固有坐標系,不進行任何坐標轉換操作
```

```
3       AXIS(num);   #如 AXIS(0.8);置區域縮放系數為 0.8
4       AXIS(bool,bool,bool,bool)  #意為 AXIS(是否轉換坐標系,縮放坐標時是否保持縱橫比,是否縮放區域,縮放區域後是否覆蓋邊界)
5       AXIS(num,bool,bool,bool)#意為 AXIS(區域縮放系數[故轉換坐標為真,縮放區域為真],縮放坐標時是否保持縱橫比,是否縮放區域[此項無效,定真],縮放區域後是否覆蓋邊界)如:
        AXIS(0.5,1,1,1);       #AXIS(0.5,1,0,1)產生相同效果
=cut
sub AXIS_test1{      #重在是否轉換 SVG 固有坐標,與區域縮放
        AXIS;#默認,可不寫.以像素為單位,Y 軸向上.
                figure;
                        line(100,100,700,400);#向上傾斜
                        #axis;#('on');#若不設區域縮放系數，則 axis()將之設定為 0.8
                figend;

        AXIS(0);#使用 SVG 固有坐標系,以像素為單位,Y 軸向下.
                figure;
                        line(100,100,700,400);#向下傾斜
                figend;
        AXIS(0.5);#區域縮放，長寬均縮至原來 0.5.
                figure;
                        line(100,100,700,400);#向上傾斜，比之第一張，縮為其一半。
                figend;

#結果應為：三個文件：SVGLAB1.svg,向上傾斜直線;
#       SVGLAB2.svg,向下傾斜直線；
#       SVGLAB3.svg,向上傾斜直線,為 SVGLAB1.svg 中之一半大小.
}
#AXIS_test1();
sub AXIS_test2{      #重在保持縱橫比縮放;與小 axis()之合作
        AXIS;#默認,可不寫.以像素為單位,Y 軸向上.
                figure;#SVGLAB1.svg
                        rect(300,300,5000,5000);#矩形畫出圖片外
                figend;

        AXIS;
                figure;#SVGLAB2.svg
                        rect(300,300,5000,5000);
                        axis;#自適應,將所畫矩形(正方形)長寬變為圖片長寬比,800:500,並分別縮放為其 0.8(axis 默認置區域縮放系數為 0.8).
                figend;
        AXIS;
                figure;#SVGLAB3.svg
                        rect(300,300,5000,5000);
                        axis;#自適應,將所畫矩形(正方形)長寬變為圖片長寬比,800:500,並分別縮放為其 0.8(axis 默認置區域縮放系數為 0.8).
                        axis(1);#保持橫縱比.
                figend;#結果:圖片中心一個正方形


        AXIS(0.8,1,1,1);#保持縱橫比縮放;區域縮放，長寬均縮至原來 0.5;
                figure;#SVGLAB4.svg
                        rect(300,300,5000,5000);
                        axis;
                figend;#結果:圖片中心一個正方形

        AXIS(0.8,0,1,1);#不保持縱橫比
                figure;#SVGLAB5.svg
                        rect(300,300,5000,5000);
                        axis;
```

```
                axis(1);          #保持縱橫比,將全局的'不保持縱橫比'覆蓋
        figend;#結果:圖片中心一個正方形

    AXIS(0.99);#區域縮放，長寬均縮至原來 0.99;
        figure;#SVGLAB6.svg
            rect(300,300,5000,5000);
            axis;#自適應,將所畫矩形(正方形)長寬變為圖片長寬比,800:500,並區域縮放系數為 0.99,即矩形幾乎佔滿整張圖片.
        figend;

#結果：見以上注釋
}
#AXIS_test2();
sub AXIS_test3{     #重在是否覆蓋縮放區域後邊界;與小 axis()之合作
        #若用 axis(x1,x2,y1,y2)設置坐標範圍，將自動覆蓋邊界。

    AXIS;  #默認,可不寫.以像素為單位,Y 軸向上.
        figure;#SVGLAB2.svg
            $x=linspace(0,6.28,100);
            $y=vsin($x);
            plot($x,$y);
            axis;#自適應,圖片位於中央
            axis(0,3.14,-1,1);#設置坐標範圍
            axis('on');
        figend;#結果:圖片中央 半周期 正弦曲線

    AXIS(1,1,1,0);#不覆蓋縮放區域後邊界
        figure;#SVGLAB3.svg
            $x=linspace(0,6.28,100);
            $y=vsin($x);
            plot($x,$y);
            axis;#自適應,圖片位於中央
            axis(0,3.14,-1,1);#設置坐標範圍
            axis('on');
        figend;#結果:圖片中央半周期餘正弦曲線,一直延伸到圖片邊界,未被覆蓋

#結果：見以上注釋
}
#AXIS_test3();
##==================AXIS 測試畢==================##
```