

第一题:

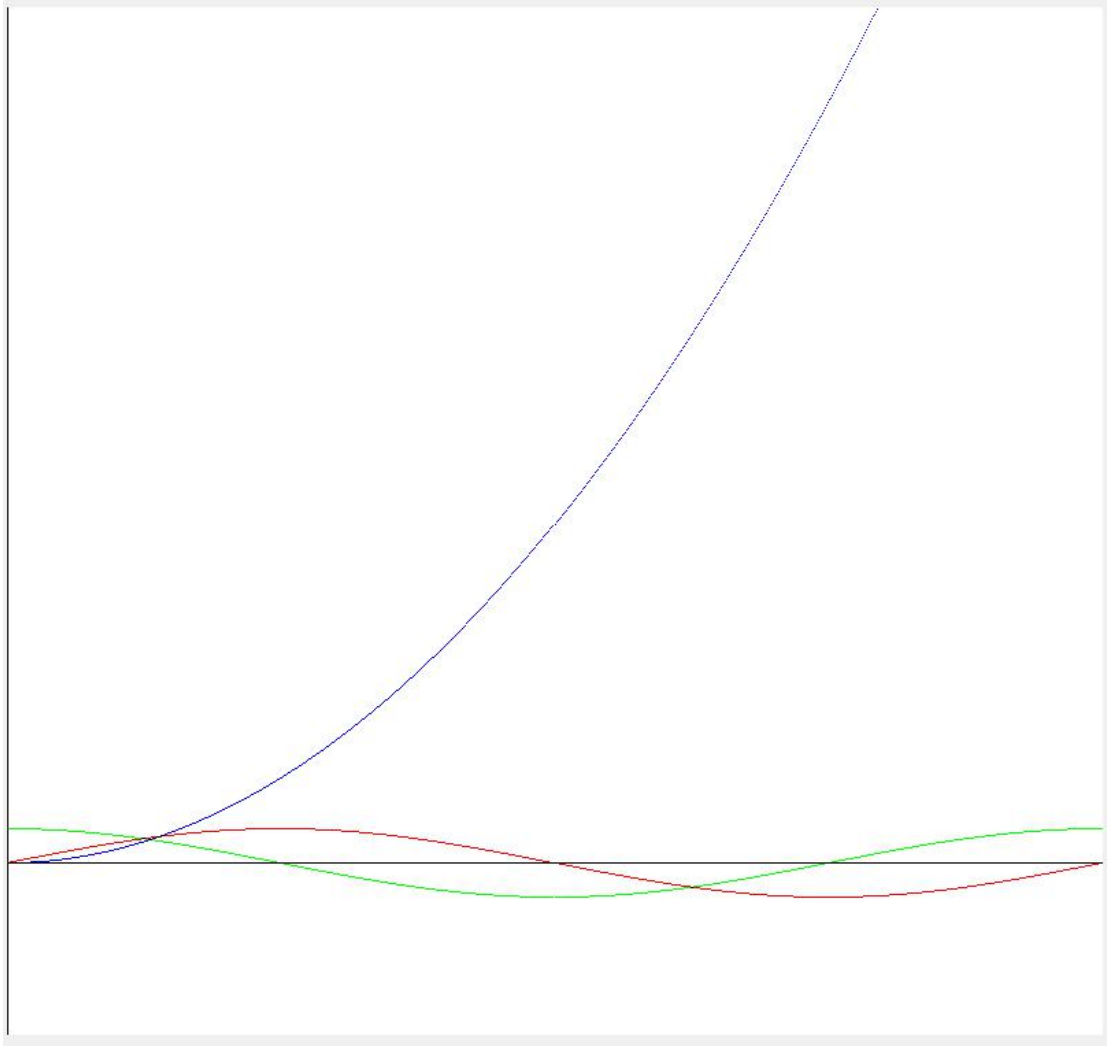
使用 **matlab** 写一个函数 **img=generateFigure (imgW,imgH)** ,其作用为产生一副彩色图像, 图像中用红色显示 $[0, 2\pi]$ 的正弦波, 用绿色显示 $[0, 2\pi]$ 的余弦波, 蓝色显示 $[0, 2\pi]$ 的  $y=x^2$  图像

将此函数编码到 **generateFigure.m** 中:

```
function[img] = generateFigure(imgH,imgW)
% 初始化
img = zeros(imgH,imgW,3);
img = uint8(img);
% 设置背景为白色
for i=1:3
    img(:, :, i)=255;
% 生成 x 在 $[0, 2\pi]$ 数据 ff
x=0:2*pi/(imgW-1):2*pi;
% 计算相应的 y
red=sin(x);
green=cos(x);
blue=x.^2;
% 把 x 映射到图片上
img_x = uint16(x/2/pi*imgW);
% 把 y 映射到图片上
img_red = uint16(imgH/30*25-round(red*imgH/30));
img_green = uint16(imgH/30*25-round(green*imgH/30));
img_blue = uint16(imgH/30*25-round(blue*imgH/30));
% 从左到右画图
for i=1:imgW
    if img_x(i)==0
        img_x(i)=img_x(i)+1;
    end
    if img_red(i)>0 && img_red(i)<=imgH
        img(img_red(i),img_x(i),2)=0;
        img(img_red(i),img_x(i),3)=0;
    end
    if img_green(i)>0 && img_green(i)<=imgH
        img(img_green(i),img_x(i),3)=0;
        img(img_green(i),img_x(i),1)=0;
    end
    if img_blue(i)>0 && img_blue(i)<=imgH
        img(img_blue(i),img_x(i),1)=0;
        img(img_blue(i),img_x(i),2)=0;
    end
end
end
```

```
% 坐标轴  
img(:,1,:)=0;  
img(imgH/30*25,:,:)=0;  
imshow(img);  
end
```

结果:



第二题:

不使用 for 循环, 实现 **bilinear interpolation**

除去调用函数, 暂时未想出解决方法, 故先用 for 循环实现, 以后再改

```
function []=bin(mag)
% magnification
img = imread('1.jpg');
% zi-oi, mag-mag
[IH,IW,ID] = size(img);
OIH = round(IH*mag); % 计算缩放后的图像高度, 最近取整
OIW = round(IW*mag); % 计算缩放后的图像宽度, 最近取整
OI = zeros(OIH,OIW,ID); % 创建新图像
% 扩展矩阵 I 边缘
IT = zeros(IH+2,IW+2,ID);
IT(2:IH+1,2:IW+1,:) = img;
IT(1,2:IW+1,:)=img(1,:,:);IT(IH+2,2:IW+1,:)=img(IH,:,:);
IT(2:IH+1,1,:)=img(:,1,:);IT(2:IH+1,IW+2,:)=img(:,IW,:);
IT(1,1,:) = img(1,1,:);IT(1,IW+2,:) = img(1,IW,:);
IT(IH+2,1,:) = img(IH,1,:);IT(IH+2,IW+2,:) = img(IH,IW,:);

% Step4 由新图像的某个像素 (zi, zj) 映射到原始图像 (ii, jj) 处, 并插值。
for zj = 1:OIW % 对图像进行按列逐元素扫描
    for zi = 1:OIH
        ii = (zi-1)/mag; jj = (zj-1)/mag;
        i = floor(ii); j = floor(jj); % 向下取整
        u = ii - i; v = jj - j;
        i = i + 1; j = j + 1;
        OI(zi,zj,:) = (1-u)*(1-v)*IT(i,j,:)
+ (1-u)*v*IT(i,j+1,:) ...
+ u*(1-v)*IT(i+1,j,:) +u*v*IT(i+1,j+1,:);
    end
end
imshow(uint8(OI));
```

效果:

