

# Geometria Computacional

Ole Peter Smith, IME, UFG, ole@mat.ufg.br

FISL11, 21/07/2010

Life sure is a Mystery to be Lived  
Not a Problem to be Solved...



- ▶ Beautiful Scientific Documents
- ▶ Classical Free Software
- ▶ Markup Language
- ▶ Thesis, automatic:
  - Bibliography
  - Contents
  - List of Tables, Figures
  - Index, References
- ▶ Slideshows: Beamer
- ▶ MikTeX, LyX, TexMaker, Kyle



- ▶ Quero Desenhar *ματεματικα...*
- ▶ Usando SL...
- ▶ Graphical Library
  - GD
  - Unices - Ruindows
  - PHP - C - Perl - ...
  - <http://www.libgd.org/Documentation>
  - [http://www.php.net/manual/pt\\_BR/book.image.php](http://www.php.net/manual/pt_BR/book.image.php)
- ▶ Formats: GIF - PNG - JPG
- ▶ ∪ Animated GIF's
- ∩ GIF's: poucas cores



- ▶ Estrutura:  
\$image=CreateImageObject;  
\$image->DrawSomething;  
...  
\$image->WriteImage;  
\$image->Close;
- ▶ Pixels... ☹



## Init Image Object (PHP)

- ▶ ImageCreate:

```
$image = imagecreate (400,400)or die ("Cannot Create image")  
#Black on White - Background first!  
$bg_color = imagecolorallocate ($image,255, 255, 255);  
$text_color = imagecolorallocate ($image,0,0,0);
```

- ▶ Tell Apache to Expect an Image

```
header("Content-type: image/png");
```

- ▶ Draw a Line

```
imageline($image,10,20,100,200,$text_color);
```

- ▶ Close and Write

```
imagepng ($image);
```



## Never Publish Emails on www

```
▶ $id=$_GET["ID"];  
$person=Mysql_SelectUniqueHash("People","ID='".$id."'");  
$email=$person[ \ "Email" ];  
header("Content-type: image/png");  
$hgt=imagefontheight(3);  
$wdt=imagefontwidth(3*strlen($email));  
$image = ImageCreate ($wdt+10, $hgt+10)  
        or die ("Cannot Create image");  
$bg_color = imagecolorallocate ($image,255, 255, 255);  
$text_color = imagecolorallocate ($image,34,139,34);  
imagecolortransparent($image,$bg_color);  
imagestring ($image,$fontsize, 5, 5, $email, $text_color);  
imagepng ($image);
```



## Draw Line

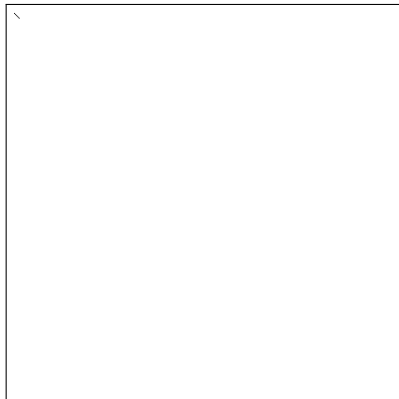
- ▶ Draw Line:  $y = x + 1, x \in [0, 5]$

```
$image = imagecreate (400,400)
           or die ("Cannot Create image");
#Black on White - Background first!
$bg_color = imagecolorallocate ($image,255, 255, 255);
$text_color = imagecolorallocate ($image,0,0,0);

header("Content-type: image/png");
imageline($image,0,1,5,6,$text_color);
imagepng ($image);
```



# Draw Line I - Result





## Draw Line I - Problem

- ▶ Escalonamento!
- ▶ Flipped: Cima por baixo...
- ▶ Draw Point: filledarc

```
void imagearc($im,  
              int $cx, int $cy,      #center  
              int $a, $int b,        #semi-eixos  
              int $ang1, int $ang2, #angle (degrees)  
              int $color);  
  
void imagefilledarc($im,  
                    int $cx, int $cy,  
                    int $a, int $b,  
                    int $ang1, int $ang2,  
                    int $color, int $style);
```

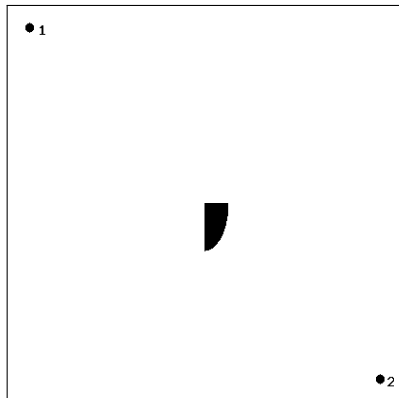


## Draw Line II

```
$im=imagecreate (400,400);  
$bgcol = imagecolorallocate ($im,255, 255, 255);  
$txtcol = imagecolorallocate ($im,0,0,0);  
header("Content-type: image/png");  
imagestring($im,5,15+8,15-5,"1",$txtcol);  
imagefilledarc($im,15,15,10,10,0,360,$txtcol,IMG_ARC_PIE);  
imagefilledarc($im,200,200,50,100,0,90,$txtcol,IMG_ARC_PIE);  
imagestring($im,5,385+8,385-5,"2",$txtcol);  
imagefilledarc($im,385,385,10,10,0,360,$txtcol,IMG_ARC_PIE);  
imagepng($im);
```



## Draw Line II - Result



## Scaling

- ▶  $R_x, R_y \in \mathbb{N}$ : Ex. 400,400
- ▶  $Min_x, Min_y \in \mathbb{R}$ : 0.0,0.0
- ▶  $Max_x, Max_y \in \mathbb{R}$ : 6.0,6.0
- ▶ Affin Application:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a_x x + b_x \\ a_y y + b_y \end{pmatrix}$$

- ▶ Conditions:

$$\begin{pmatrix} 0 \\ R_y \end{pmatrix} = \begin{pmatrix} a_x Min_x + b_x \\ a_y Min_y + b_y \end{pmatrix}, \quad \begin{pmatrix} R_x \\ 0 \end{pmatrix} = \begin{pmatrix} a_x Max_x + b_x \\ a_y Max_y + b_y \end{pmatrix}$$



## Scaling

- ▶  $a_x(Max_x - Min_x) = R_x - 0 \wedge a_y(Max_y - Min_y) = 0 - R_y \Leftrightarrow$
- ▶  $a_x = \frac{R_x}{Max_x - Min_x} \wedge a_y = -\frac{R_y}{Max_y - Min_y}$
- ▶  $b_x = -a_x Min_x$
- ▶  $b_y = -a_y Max_y$



## Scaling - Init

```
▶ function InitCanvas($R,$Min,$Max)
{
    global $a,$b;
    $a=array();$b=array();

    $a[0]=$R[0]/($Max[0]-$Min[0]);
    $a[1]=-$R[1]/($Max[1]-$Min[1]);

    $b[0]=-$a[0]*$Min[0];
    $b[1]=$a[1]*$Max[1];
}
```



## Scaling - Conversion

```
▶ function ScalePoint($p)
{
    global $a,$b;

    $pp=array();
    for ($i=0;$i<2;$i++)
    {
        $pp[$i]=((int) ($a[$i]*$p[$i]+$b[$i]));
    }

    return $pp;
}
```



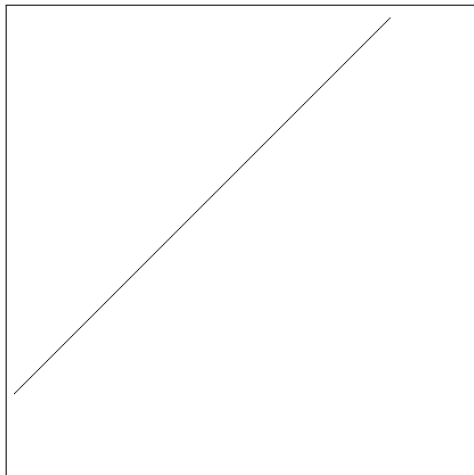
## Now Draw the Line....

```
▶ $Res=array(400,400);  
$Min=array(0.0,0.0);  
$Max=array(6.0,6.0);  
InitCanvas($R,$Min,$Max);  
$p1=array(0.0,1.0);  
$p2=array(5.0,6.0);  
$pp1=ScalePoint($p1);  
$pp2=ScalePoint($p2);  
imageline($pp1[0],$pp1[1],$pp2[0],$pp2[1],$color);
```





## Draw Line I - Result



## Serial Conversion

- ▶ Maligma: Código Repetido

- ▶ `function ScalePoints($ps)`

```
{  
    $pps=array();  
    for ($n=0;$n<count($ps);$n++)  
    {  
        $pps[$n]=ScalePoint($ps[$n]);  
    }  
  
    return $pps;  
}
```



# Drawing a Circle

- ▶ Equation:

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

- ▶ Parametrization:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_c + r \cos t \\ y_c + r \sin t \end{pmatrix}$$

- ▶  $t \in [-\pi, \pi]$



## Generating a Circle

```
▶ function Circle($n,$pc,$r,$ang=0.0)
{
    $dt=2*3.1415927/(1.0*($n-1));
    $ps=array();
    for ($i=0,$t=$ang;$i<$n;$i++,$t+=$dt)
    {
        $p=array( $pc[0]+$r*cos($t),$pc[1]+$r*sin($t) );
        array_push($ps,$p);
    }
    return $ps;
}
```



## Curve as Line Segments

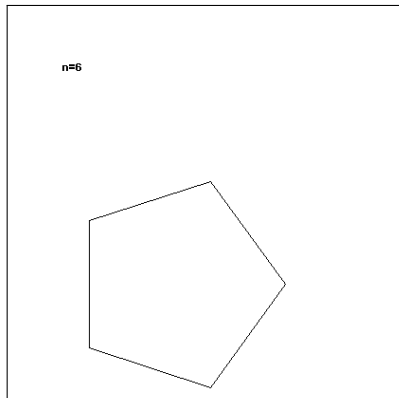
```
▶ function DrawCurve($im,$ps,$color)
{
    $pps=ScalePoints($ps);
    for ($n=0;$n<count($pps)-1;$n++)
    {
        imageline($im,$pps[$n][0],$pps[$n][1],
                $pps[$n+1][0],$pps[$n+1][1],$color);
    }
}
```

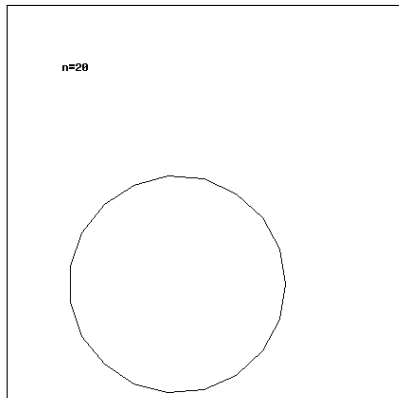


# Drawing Circle

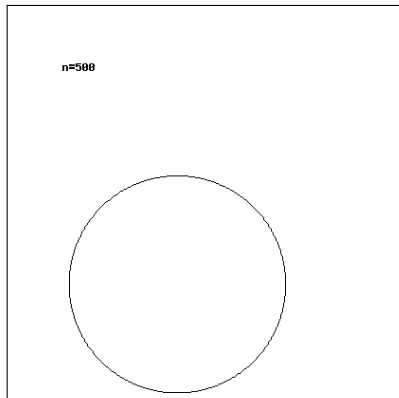
```
▶ $n=6;  
$center=array(2.0,1.0);  
$ps=Circle($n,$center,2);  
DrawCurve($im,$ps,$color);  
imagestring($im,3,5,5,"n=$n",$color);
```



Draw Circle - Result  $n = 6$ 

Draw Circle - Result  $n = 20$ 



Draw Circle - Result  $n = 500$ 

## Drawing a Wheel

- ▶ 

```
function DrawWheel($im,$n,$dn,$pc,$r1,$r2,$color,$ang=0)
{
    $p1s=Circle($n,$pc,$r1,$ang);
    DrawCurve($im,$p1s,$color);
    $p2s=Circle($n,$pc,$r2,$ang);
    DrawCurve($im,$p2s,$color);

    for ($i=0;$i<$n;$i+=$dn)
    {
        $p1=ScalePoint($p1s[$i]);
        $p2=ScalePoint($p2s[$i]);
        imageline($im,$p1[0],$p1[1],$p2[0],$p2[1],$color);
    }
}
```

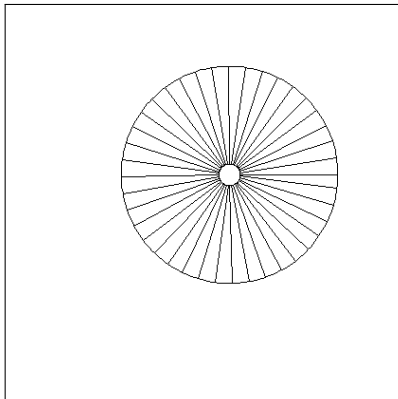


## Drawing a Wheel

```
▶ $n=200;  
$dn=5;  
$center=array(2.0,1.0);  
DrawWheel($im,$n,$dn,$center,$color,3.0,0.2);
```



# Wheel - Result



## Animate Wheel

```
▶ $nimages=79;  
$dt=2.0*3.1415927/$nimages;  
system("/bin/rm Wheel*.gif");  
for ($t=0.0,$i=0;$i<$nimages;$i++)  
{  
    $image = ImageCreate...  
    $center=array(3.0,3.0);  
    DrawWheel($image,200,5,$center,0.2,2.0,$color,$t);  
    imagegif($image,"Wheel.$i.gif");  
    $t+=$dt;  
}
```

//Merda! PHP nao tem suporte para GIF animado



## Animate Wheel

- ▶ ImageMagick...
- ▶ Continued...

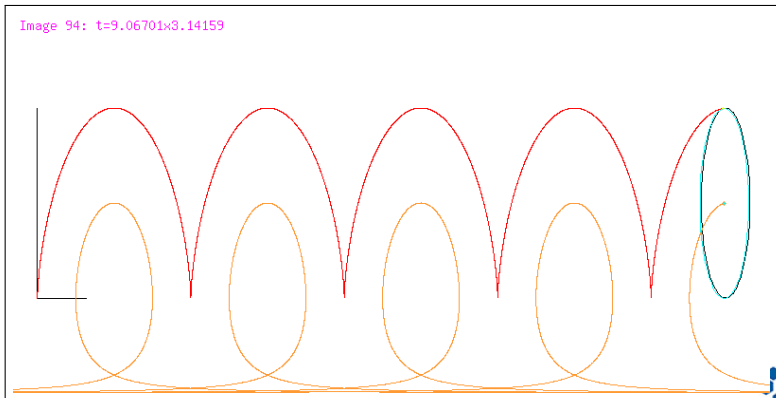
```
system("/usr/bin/convert -delay 2 -loop 0 "  
        "Wheel.*.gif Wheel.gif");  
header("Content-type: text/html");  
print "<IMG SRC='Wheel.gif'>\n";
```

- ▶ <http://127.0.0.1/html/uruacu/6.php>



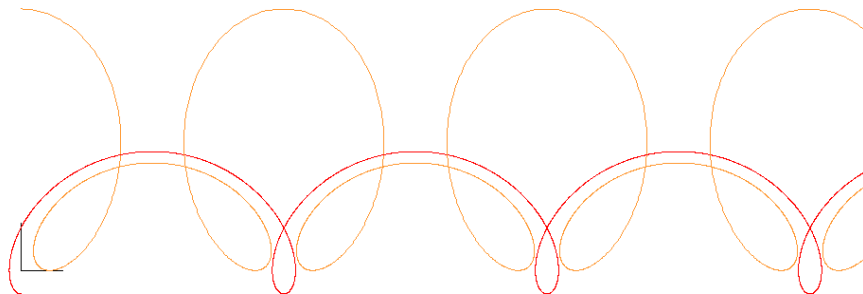
# Cycloid

Image 94:  $t=9.06701 \times 3.14159$



# Trochoid

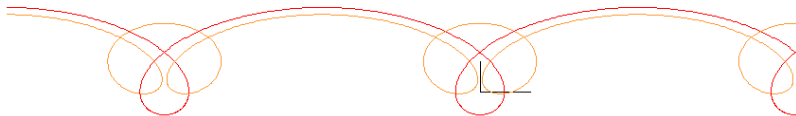
Image 45:  $t=7.18367 \times 3.14159$



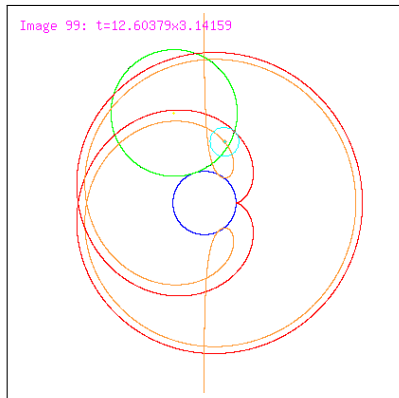


# Trochoid

Image 62:  $b/a=0.76633$

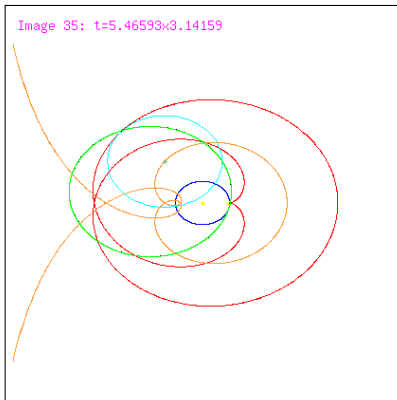


# Epicycloid



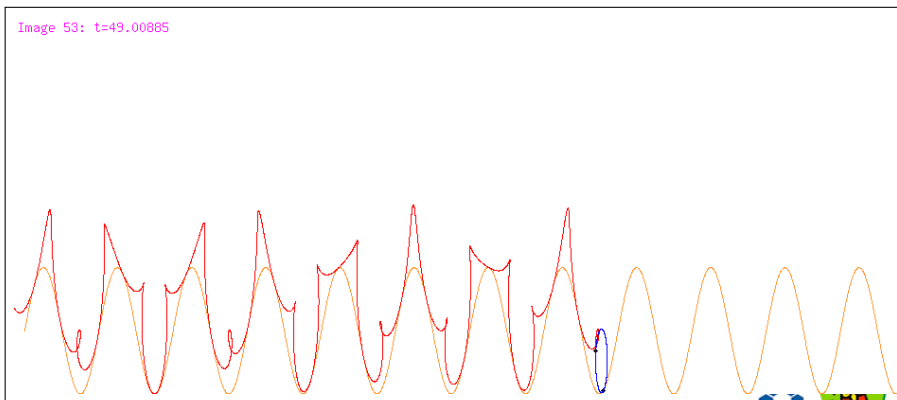
# Hypocycloid

Image 35:  $t=5.46593 \times 3.14159$



# Oloid

Image 53: t=49.00885



Please Always Enjoy... [Animations - Made in C++](#)

