Geometria Computacional 2D

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

EMSL '10
15/10/2010/2010
UFU, Uberlândia-MG

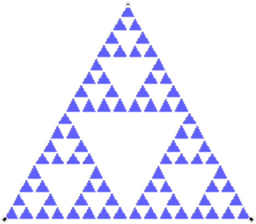
The Road to Knowledge? Simple:
Err, and Err, and Err again
But Less, and Less, and Less
Piet Hein

Real programmers can write assembly code in any language
Larry Wall

TIMTOWTDI: There is more than one Way to Do It!
Larry Wall

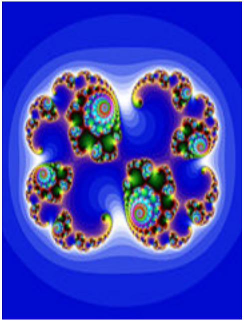
Curiosity

- Inteligência: Potencial
- Sabedoria: Disciplina & Curiosidade
- Graphics
- Programming
- Math & Art
- Visualization - Animation
- The Road to Wisdom? Simple, my Friend:
  Err, Err and Err Again...
  But Less and Less and Less
  Piet Hein
- Programming vs. Medicine...



The Mathematics

- Geometria Analítica
  2D - 3D
  Planos - Curvas - Superfícies
  Euclidian
- Álgebra Linear
  Matrizes - Transformações - AutoVetor/Valor
- Geometria Diferencial
  Cinemática
- Análise Numérica
  Ferramenta
- Geometria Projetiva
  Noneuclidian
- Fractais



Implementation

- Language: Perl, PHP, C/C++, Java,....
- Graphics Library:
  GD - Others
- **Ex:** Apache - HTML - PHP - Browser
- www.apache.org
- www.php.org - www.perl.org
- www.gd.org
- O mais Importante não é Saber de Tudo
  O mais Importante Saber onde Procurar!
- Limite da Minha Pesquisa Pessoal:
  Minha Própria
  Inteligência, Disciplina e Curiosidade

- Estrutura:

```
$image=CreateImageObject;
$image->DrawSomething;
$image->imageline(10,10,390,390);
$image->imagearcfilled(200,200,10,10);
$image->WriteImage;
$image->Close;
```

- Pixels - Coordenadas

Image

- Formats: PNG - GIF - JPG - ...
- Animated GIF - MPG
- Resolution: $(X, Y) = (800x600)$
- Physical: $[x_{min}, x_{max}]x[y_{min}, y_{max}]$
  Canonical: $[0, 1]x[0, 1]$
- $\frown (X, Y) = (0, 0)$ é Upper Left
- Linear Transformation:

$$\underline{\mathbf{X}} = \underline{\underline{\mathbf{A}}}\ \underline{\mathbf{x}} \Leftrightarrow \underline{\mathbf{x}} = \underline{\underline{\mathbf{A}}}^{-1}\ \underline{\mathbf{X}}$$

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \underline{\underline{\mathbf{A}}}\ \underline{\mathbf{x}} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{pmatrix}$$

- $$\underline{\underline{\mathbf{A}}} \begin{pmatrix} x_{min} \\ y_{min} \end{pmatrix} = \begin{pmatrix} 0 \\ Y \end{pmatrix}, \quad \underline{\underline{\mathbf{A}}} \begin{pmatrix} x_{max} \\ y_{max} \end{pmatrix} = \begin{pmatrix} X \\ 0 \end{pmatrix}$$

Image

- $$\underline{\underline{\mathbf{A}}}\begin{pmatrix} x_{min} & y_{min} \\ x_{max} & y_{max} \end{pmatrix} = \begin{pmatrix} 0 & X \\ Y & 0 \end{pmatrix} \Leftrightarrow \underline{\underline{\mathbf{A}}} = \begin{pmatrix} 0 & X \\ Y & 0 \end{pmatrix} \underline{\underline{\mathbf{X}}}^{-1}$$
- **Tarefa**: Inversa de Matriz *Regular* (det $\neq 0$) - Gauss - LU
- **Tarefa**: Imagem de Aplicação Linear:

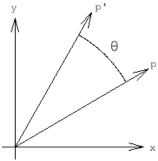
```
for ($i=0;$i<$DIM;$i++)
{
   $X[$i]=0.0;
   for ($j=0;$j<$DIM;$j++)
   {
       $X[$i]+=$A[$i][$j]*$x[$j];
   }
   $X[i]=$sum
}
```

Dot Product

- Measure: Lenths and Angles
- $\underline{\mathbf{x}} = (x_1, ..., x_n)$, $\underline{\mathbf{y}} = (y_1, ..., y_n)$

$$\underline{\mathbf{x}} \cdot \underline{\mathbf{y}} = x_1 y_1 + ... + x_n y_n = \sum_{i=1}^{n} x_i y_i = \underline{\mathbf{x}}^T \underline{\mathbf{y}}$$

```
$dot=0.0);
for ($i=0;$i<$N;$i++)
{
    $dot+=$x[$i]*$y[$i];
}
```

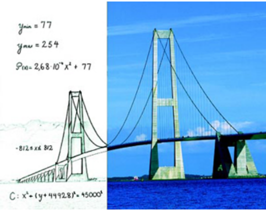


- Length: $|\underline{\mathbf{x}}| = \sqrt{\underline{\mathbf{x}} \cdot \underline{\mathbf{x}}}$
- Angle: $\cos \angle(\underline{\mathbf{a}}, \underline{\mathbf{b}}) = \frac{\underline{\mathbf{a}} \cdot \underline{\mathbf{b}}}{|\underline{\mathbf{a}}||\underline{\mathbf{b}}|}$

Plane Curves

- Equation: $F(x,y)=0$
- Function: $\underline{\mathbf{r}}(x)=\begin{pmatrix} x \\ f(x) \end{pmatrix}$
- Parametrization:

$$\underline{\mathbf{r}}(t)=\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \quad t_1 \le t \le t_2$$




```
$dt=($t2-$t1)/($N-1);
for ($i=0,$t=$t1;$i<$N;$i++)
{
    array_push($ps,array(x($t),y($t)));
}
```

- Building Stones: Line Segments - ArcFilled

Line

- Line Segment: $\underline{\mathbf{p}}_1 = (x_1, y_2)$ e $\underline{\mathbf{p}}_2 = (x_2, y_2)$.
  $\underline{\mathbf{r}} = \underline{\mathbf{p}}_2 - \underline{\mathbf{p}}_1$.
- Versor de $\underline{\mathbf{r}} = (r_x, r_y)$: $\underline{\mathbf{n}} = \widehat{\underline{\mathbf{r}}} = (-r_y, r_x)$
- Parametrization/Convex Combination:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + t \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}, \quad 0 \leq t \leq 1$$

- Normal:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -r_y \\ r_x \end{pmatrix}$$

- Equation:

$$a(x - x_1) + b(y - y_1) = 0 \Leftrightarrow ax + by = ax_1 + by_1 = c$$

Circle/Ellipse

- Center: $\underline{\mathbf{p}}_c = (x_c, y_c)$, half-axes: a, b
- $a = b$: Circle
- Equation:
$$\left(\frac{x - x_c}{a}\right)^2 + \left(\frac{y - y_c}{b}\right)^2 = 1$$
- Parametrization:
$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_c + a\cos t \\ y_c + b\sin t \end{pmatrix} \quad 0 \leq t \leq 2\pi$$

Hiperbola

- Center: $\underline{\mathbf{p}}_c = (x_c, y_c)$ e halfaxes, a, b
- Equation:

$$\left(\frac{x - x_c}{a}\right)^2 - \left(\frac{y - y_c}{b}\right)^2 = \pm 1$$

- Parametrization:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_c + a\cosh t \\ y_c + b\sinh t \end{pmatrix} \quad -\alpha \leq t \leq \alpha$$

- Funções Hiperbólicas

$$\cosh t = \frac{e^t + e^{-t}}{2} \qquad \sinh t = \frac{e^t - e^{-t}}{2}$$

$$\cosh^2 t - \sinh^2 t = 1$$

Cinematics

- Position:

$$\underline{\mathbf{r}}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

- Tangent & Velocity:

$$\underline{\mathbf{r}}'(t) = \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} \quad v(t) = \sqrt{\underline{\mathbf{r}}'(t) \cdot \underline{\mathbf{r}}'(t)} = \frac{1}{\sqrt{x'(t)^2 + y'(t)^2}}$$

- Unit Tangent & Normal:

$$\underline{\mathbf{t}}(t) = \frac{1}{v(t)} \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix}, \qquad \underline{\mathbf{n}}(t) = \widehat{\underline{\mathbf{t}}}(t) = \frac{1}{v(t)} \begin{pmatrix} -y'(t) \\ x'(t) \end{pmatrix},$$

- Acceleration:

$$\underline{\mathbf{r}}''(t) = \begin{pmatrix} x''(t) \\ y''(t) \end{pmatrix}$$

- Arc Length: $s(t) - s(t_0) = \int_{t_0}^{t} \sqrt{x'(t)^2 + y'(t)^2} dt$

Curvature

- Oscillating Circe: Same tangent & Accelleration

$$\rho = \frac{[\underline{\mathbf{r}}'(t)\underline{\mathbf{r}}''(t)]}{\sqrt{x'(t)^2 + y'(t)^2}^3}$$

- Radius of Oscillating Circe, ρ:

$$\rho(t) = \frac{[\underline{\mathbf{r}}'(t)\underline{\mathbf{r}}''(t)]}{\sqrt{x'(t)^2 + y'(t)^2}^3}$$

- Curvature:

$$\kappa(t) = \frac{1}{\rho(t)}$$

- Curvature Vector & Center of Curvature
- Curvature:

$$\underline{\mathbf{p}}_c(t) = \underline{\mathbf{r}}(t) + \rho(t)\underline{\mathbf{n}}(t)$$

Transformations

- Linear:

$$f(\underline{x}) = \underline{\underline{A}}\underline{x},$$
$$f(\alpha\underline{x} + \beta\underline{y}) = \alpha f(\underline{x}) + \beta f(\underline{y})$$

If $\det \underline{\underline{A}} \neq 0$:

$$f^{-1}(\underline{y}) = \underline{\underline{A}}^{-1}\underline{y}$$

- Affin:

$$f(\underline{x}) = \underline{\underline{A}}\underline{x} + \underline{b}$$

- Rotate Coordinate System:

$$\underline{x}' = \underline{\underline{D}}\underline{x}, \qquad \underline{\underline{D}}^{-1} = \underline{\underline{D}}^{T}$$

$\underline{\underline{D}}$ ortogonal: Preserves Lengths and Angles:

$$(\underline{\underline{D}}\,\underline{x}) \cdot (\underline{\underline{D}}\,\underline{x}) = \underline{x} \cdot \underline{x}$$

Scaling

- Scaling resp. Origin, factor $\lambda \in \mathbb{R}$ around x-axis:

$$\underline{\mathbf{x}}' = \begin{pmatrix} \lambda & 0 \\ 0 & 1 \end{pmatrix} \underline{\mathbf{x}} = \begin{pmatrix} \lambda x \\ y \end{pmatrix} = \underline{\underline{\mathbf{L}}}(\lambda)\ \underline{\mathbf{x}}$$

$\lambda = 1$:	Identity, $\underline{\underline{\mathbf{I}}}$	
$\lambda = -1$:	Reflection in y axis	<– Rev. Orient.
$\lambda = 0$:	Projection onto y axis	
$\lambda > 1$:	Expansion	
$0 < \lambda < 1$:	Dilatation	

- Bidirectional Scaling:

$$\underline{\mathbf{x}}' = \begin{pmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{pmatrix} \underline{\mathbf{x}} = \begin{pmatrix} \lambda_x x \\ \lambda_y y \end{pmatrix} = \underline{\underline{\mathbf{L}}}(\lambda_x, \lambda_y)\ \underline{\mathbf{x}}$$

- $$\begin{pmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{pmatrix} = \begin{pmatrix} \lambda_x & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \lambda_y \end{pmatrix}$$

UFG

Translation & Rotation

- Translation, $\underline{\mathbf{a}}$ - $\frown$ Não Linear:

$$\underline{\mathbf{x}}' = \underline{\mathbf{t}} + \underline{\mathbf{x}}$$

- Rotation resp. Origin, Angle θ - $\smile$ Linear:

$$\underline{\mathbf{x}}' = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \underline{\mathbf{x}} = \underline{\underline{\mathbf{D}}}(\theta)\ \underline{\mathbf{x}},$$

$$\underline{\underline{\mathbf{D}}}(\theta)^{-1} = \underline{\underline{\mathbf{D}}}(-\theta) = \underline{\underline{\mathbf{D}}}(\theta)^{T}$$

Projetive Geometry

- Camera at Origin, Looking at $(0, 0, 1)$
- Translation in *non-linear*...
  That Really Sucks!
- Remedy: Drawing plane is $z = 1$ (instead of $z = 0$).
- Point in $\mathbb{R}^2$: (x, y).
  Point in $\mathbb{R}^3$: $(x, y, 1)$.
- Do Something to Point
- Point in $\mathbb{R}^3$: (x, y, z).
  Point in $\mathbb{R}^2$, $z \neq 0$: $(x/z, y/z)$.
  Point in Infinity: $z = 0$: $(x, y)_\infty$.

Projetive, Translation

- Translation:

$$\underline{\underline{T}}(\underline{t}) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

Inverse:

$$\begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x - t_x \\ y - t_y \\ 1 \end{pmatrix}$$

- $$\underline{\underline{T}}(\underline{t})^{-1} = \underline{\underline{T}}(-\underline{t})$$

Projetive, Rotation

- Rotation:

$$\underline{\underline{\mathbf{R}}}(\theta)\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\underline{\underline{\mathbf{R}}}(\theta)^{-1} = \underline{\underline{\mathbf{R}}}(-\theta) = \underline{\underline{\mathbf{R}}}(\theta)^T$$

- Rotation, Angle θ, Resp. to $\underline{\mathbf{p}} = (p_x, p_y)$:
  1: Translation to $\underline{\mathbf{p}}$: $\underline{\underline{\mathbf{T}}}(\underline{\mathbf{p}})$
  2: Rotation by (new) Origin: $\underline{\underline{\mathbf{R}}}(\theta)$
  3: Translation back: $-\underline{\mathbf{p}}$ $\underline{\underline{\mathbf{T}}}(-\underline{\mathbf{p}}) = \underline{\underline{\mathbf{T}}}(\underline{\mathbf{p}})^{-1}$
- Apply from Right:

$$\underline{\underline{\mathbf{R}}}'(\theta) = \underline{\underline{\mathbf{T}}}(\underline{\mathbf{p}})^{-1}\underline{\underline{\mathbf{R}}}(\theta)\underline{\underline{\mathbf{T}}}(\underline{\mathbf{p}})$$

- Similar Matrices: $\underline{\underline{\mathbf{B}}} = \underline{\underline{\mathbf{D}}}^{-1}\underline{\underline{\mathbf{A}}}\underline{\underline{\mathbf{D}}}$
  Describe same Transformation in Different Bases

Scaling

- Unidirectional Scaling with Resp. to y-axis:

$$\begin{pmatrix} \lambda & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Unidirectional Scaling with Resp. to $\underline{e} = (\cos\theta, \sin\theta)$:
- Coordinate System: $\underline{e}, \underline{\widehat{e}}$

$$\underline{\underline{D}}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

$$\underline{x}_{old} = \underline{\underline{D}}(\theta)\underline{x}_{new} \Leftrightarrow \underline{x}_{new} = \underline{\underline{D}}(\theta)^{-1}\underline{x}_{old}$$

- 1: Change Coordinate System:

$$\underline{\underline{R}}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Scaling

- 2: Scale with Resp. to y-axis

$$\underline{\underline{\mathbf{L}}}(\lambda) = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- 3: Rotate back

$$\underline{\underline{\mathbf{R}}}(-\theta) = \underline{\underline{\mathbf{R}}}(\theta)^{-1} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Composition from right:

$$\underline{\underline{\mathbf{R}}}(\theta)^{-1}\underline{\underline{\mathbf{L}}}(\lambda)\underline{\underline{\mathbf{R}}}(\theta)$$

Again: Similar Matrices

Animation

```
$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...
system("/usr/bin/convert -delay 2 -loop 0 ".
       "Wheel.*.gif Wheel.gif");
```

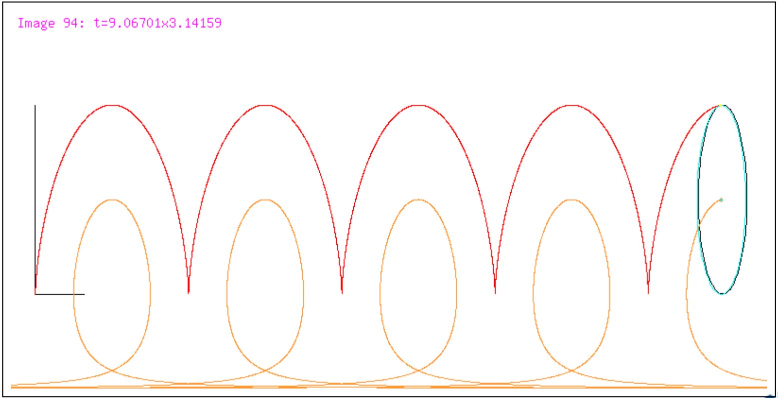
Rolling

- Given: Fixed Curve $X(t), Y(t), 0 \leq t \leq 1$.
- Circle to Roll, Radius r
  Mark Point
- Initially: Point in (X_0, Y_0).
- Roll Circle without Sliding:

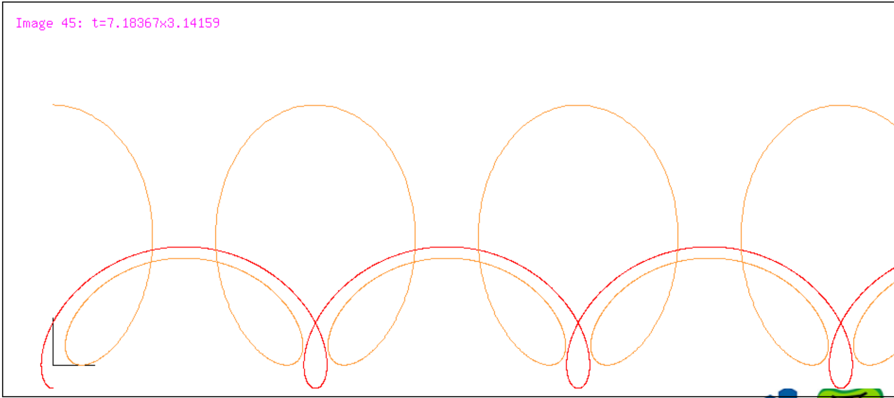
$$s(t) = r\theta$$

- Cycloid: Circle Rolling on Line
- Epicycloid: Circle Outside Circle
- Hypocycloid: Circle Inside Circle
- Trochoid, Epitrochoid, Hypotrochoid,...
- Oloid

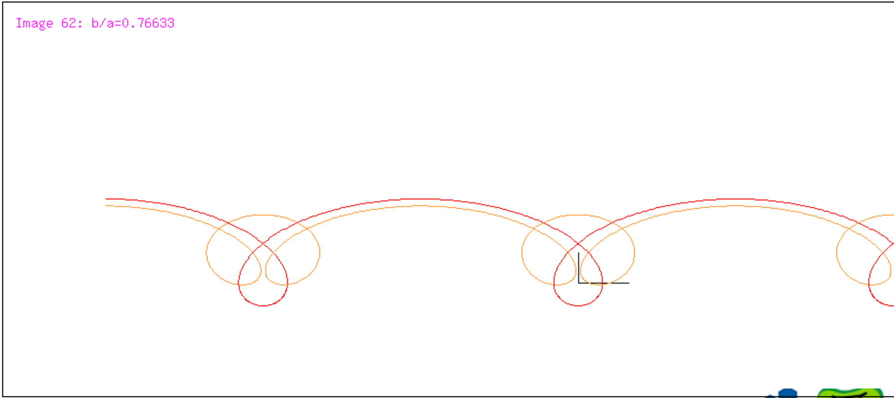
Cycloid



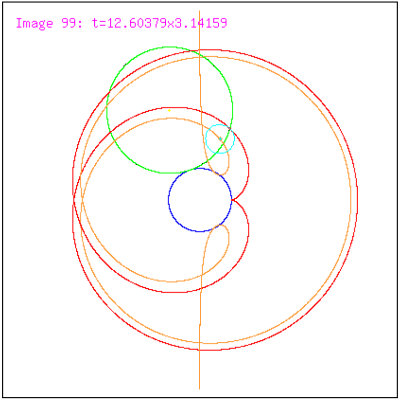
Trochoid



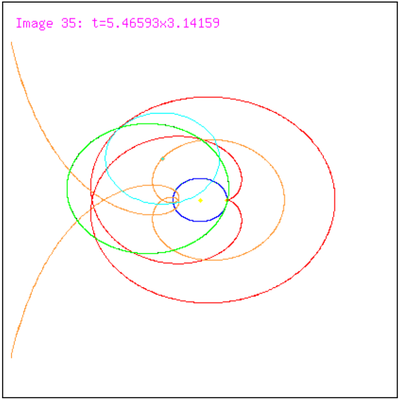
Trochoid



Epicycloid



Hypocycloid



Oloid