Curvature and torsion for implicitely defined curves

Plane Curves

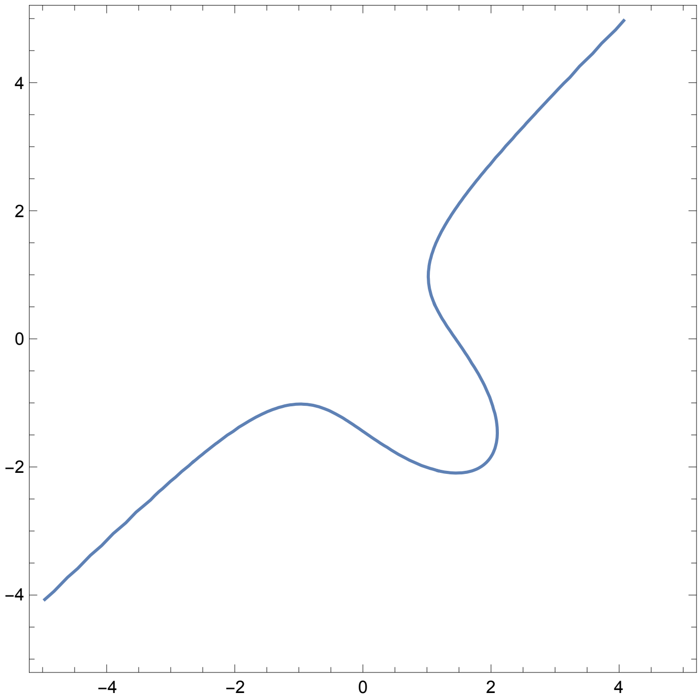
Suppose we have a curve given by an equation:

$$f(x, y) = 0$$

Here is an algebraic example:

```
f[x_, y_] := x^3 + 3 x y - y^3 - 3
```

```
ContourPlot[f[x, y] == 0, {x, -5, 5}, {y, -5, 5}]
```



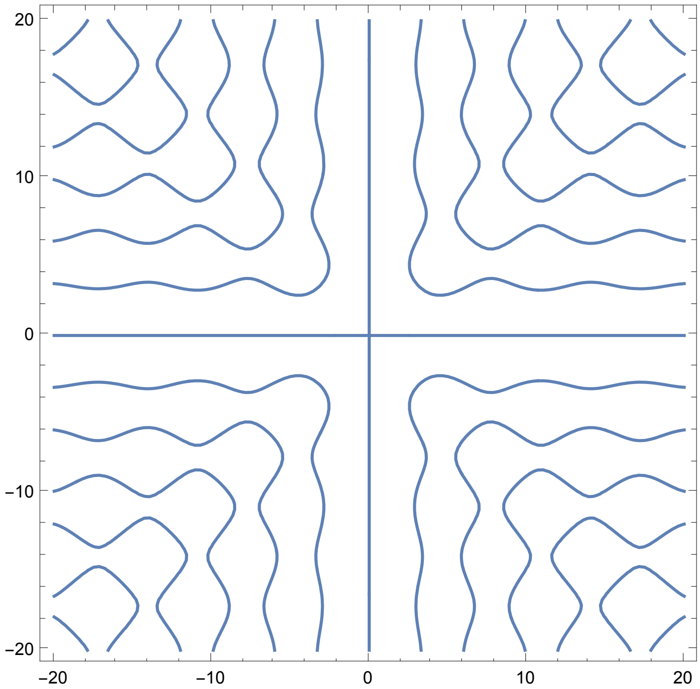
Here is a non-algebraic example

```
g[x_, y_] := x Sin[y] + y Sin[x]
```

```
∇{x,y} g[x, y]
```

```
{y Cos[x] + Sin[y], x Cos[y] + Sin[x]}
```

```
ContourPlot[x Sin[y] + y Sin[x] == 0, {x, -20, 20}, {y, -20, 20}]
```



The question is : how to compute the curvature of a curve given by this kind of equation at a point (a,b) lying on the curve?

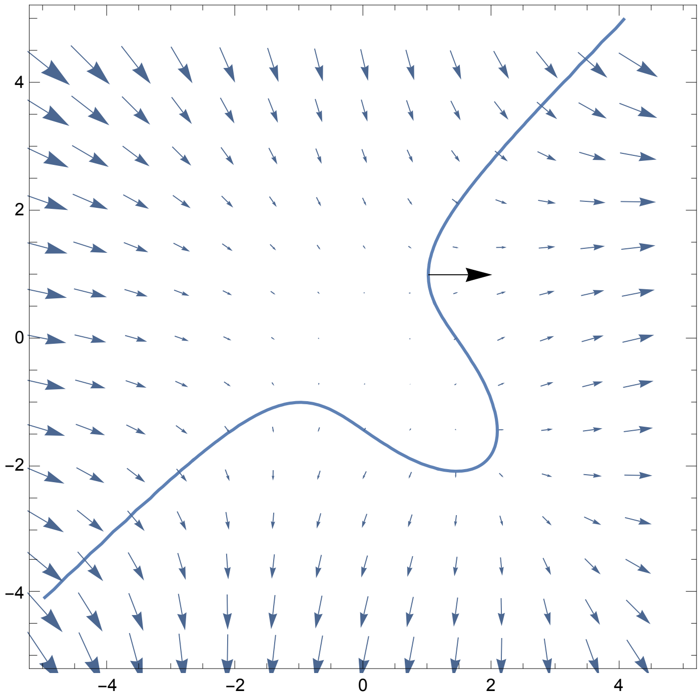
The crucial observation is that we can find the Frenet-Serret system without the need for a parametric representation. The first observation is that it is very easy to find a unit normal and unit tangent fields.

Normal vector

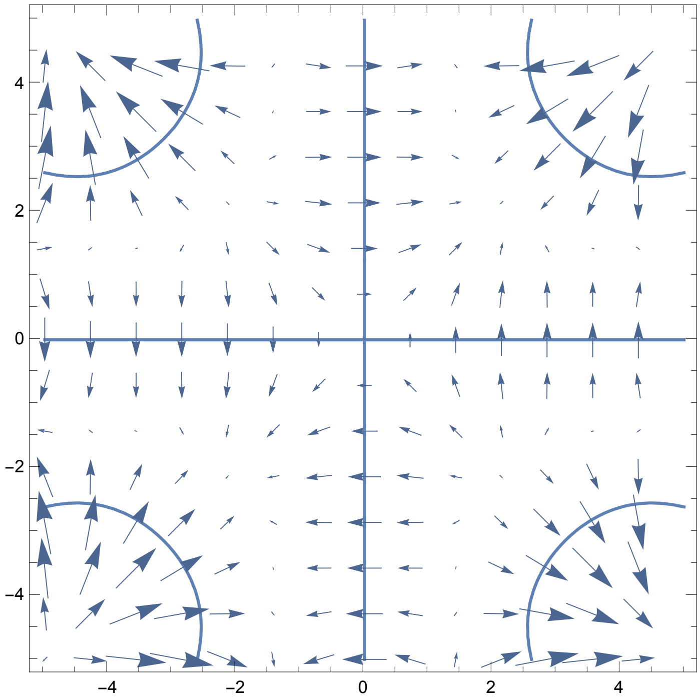
```
nr[x_, y_] = ∇{x,y} f[x, y]
```

$\{3 x^2 + 3 y, 3 x - 3 y^2\}$

```
Show[{ContourPlot[f[x, y] == 0, {x, -5, 5}, {y, -5, 5}],
  Graphics[Arrow[{{1, 1}, {1, 1} + Normalize[nr[1, 1]]}]],
  VectorPlot[{3 x^2 + 3 y, 3 x - 3 y^2}, {x, -5, 5}, {y, -5, 5}]}]
```




```
Show[{ContourPlot[g[x, y] == 0, {x, -5, 5}, {y, -5, 5}],
  VectorPlot[{y Cos[x] + Sin[y], x Cos[y] + Sin[x]}, {x, -5, 5}, {y, -5, 5}]}]
```

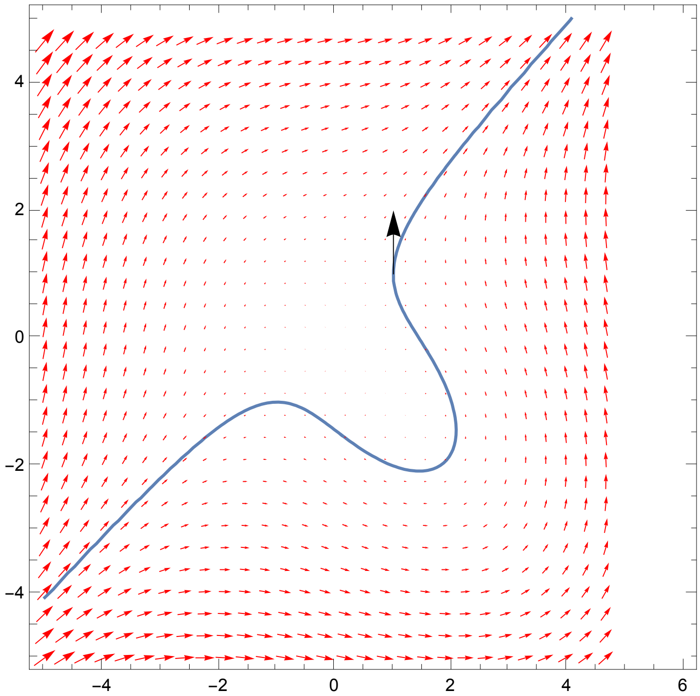



```
tg[x_, y_] := RotationTransform[Pi / 2][nr[x, y]]
```

```
tg[x, y]
```

$\{-3x + 3y^2, 3x^2 + 3y\}$

```
Show[{ContourPlot[f[x, y] == 0, {x, -5, 6}, {y, -5, 5}],
  Graphics[Arrow[{{1, 1}, {1, 1} + Normalize[tg[1, 1]]}]],
  VectorPlot[{-3 x + 3 y^2, 3 x^2 + 3 y}, {x, -5, 5}, {y, -5, 5},
   VectorColorFunction → (Red &), VectorPoints → 30]}]
```



Now we can obtain the formula for curvature using the same method as for a parametric description

$$k(f\_, \{x\_, y\_\}) := -\frac{-2 \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \frac{\partial^2 f}{\partial x \partial y} + \frac{\partial^2 f}{\partial x^2} \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2 \frac{\partial^2 f}{\partial y^2}}{\left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)^{3/2}}$$

```
k[f[x, y], {x, y}] /. {x → 1, y → 1}
```

1

$$\kappa_s[\{x\_, y\_\}][t\_] = \frac{-y'[t]\, x''[t] + x'[t]\, y''[t]}{\left(x'[t]^2 + y'[t]^2\right)^{3/2}};$$

One can also solve the problem numerically by constructing an approximate local parametri - sation

```
D[f[x, y[x]], x]
```

$3 x^2 + 3 y[x] + 3 x y'[x] - 3 y[x]^2 y'[x]$

We need to find some point on the curve, e.g. let's take $x = 0$:

```
p = First[y /. Solve[f[0, y] == 0, y, Reals]]
```

Root[3 + #1^3 &, 1]

```
Clear[y]
```

```
h = y /. NDSolve[{3 x^2 + 3 y[x] + 3 x y'[x] - 3 y[x]^2 y'[x] == 0, y[0] == p},
    y, {x, -0.5, 0.5}, WorkingPrecision → 20][[1]]
```

InterpolatingFunction[Domain: {{-0.50000000000000000000, 0.50000000000000000000}} Output: scalar]

```
h[0]
```

-1.4422495703074083823

```
h[0]
```

-1.44225

```
f[0, h[0]] // N
```

0.

```
gr1 = Plot[h[x], {x, -0.5, 0.5}, PlotStyle → {Green, Dashing[0.2]}];
```

```
gr2 = ContourPlot[f[x, y] == 0, {x, -0.5, 0.5},
   {y, p - 1, p + 1}, Axes → True, ColorFunction → Function[{x, y}, Red]];
```

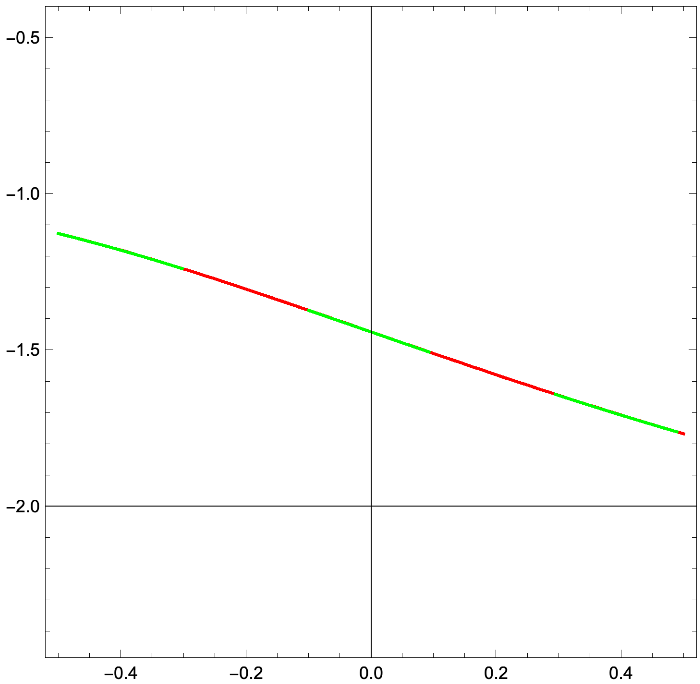
```
k[f[x, y], {x, y}] /. {x → 0, y → p}
```

0

```
κS[{# &, h[#] &}][0]
```

-7.04×10^{-6}

```
Show[gr2, gr1]
```



Drawing plane curves with Assigned Curvature

Fundamental Theorem of Plane Curves

Let α and γ be unit curves definded on the same interval (a, b). Suppose α and β have the same signed curvature. Then there is an orientation preserving Euclidean motion mapping α into γ. Given a piecwise-continuous function $\kappa : (a, b) \to \mathbb{R}$, a unit curve with parametric equation β is given by

$$\theta(s) = \int \kappa(s)\, ds + s_0$$

$$\beta(s) = \left\{c + \int \cos(\theta(s))\, ds,\, d + \int \sin(\theta(s))\, ds\right\}$$

We can use these formulas to find (numerically) a curve with any given curvature:

```
intrinsic[fun_, a_: 0, {c_: 0, d_: 0, θ_: 0}, {smin_: - 10, smax_: 10}, opts___] :=
 Module[{x, y, th}, Flatten[{x, y} /.
    NDSolve[{x'[ss] == Cos[th[ss]], y'[ss] == Sin[th[ss]], th'[ss] == fun[ss],
      x[a] == c, y[a] == d, th[a] == θ}, {x, y, th}, {ss, smin, smax}, opts]]]

intrinsic[(# + Sin[#]) &, 0, {0, 0, 0}, {-18, 18}]
```

```
{InterpolatingFunction[ Domain: {{-18., 18.}} Output: scalar ],
 InterpolatingFunction[ Domain: {{-18., 18.}} Output: scalar ]}
```

```
Clear[f]

f[t_] = Through[intrinsic[(# + Sin[#]) &, 0, {0, 0, 0}, {-18, 18}][t]]
```

```
{InterpolatingFunction[ Domain: {{-18., 18.}} Output: scalar ][t],
 InterpolatingFunction[ Domain: {{-18., 18.}} Output: scalar ][t]}
```

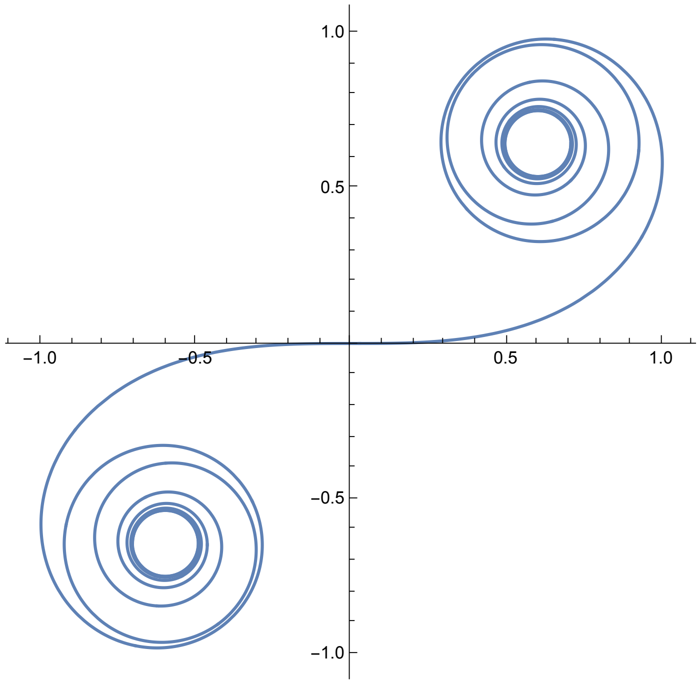
```
f[1.5]
```

```
{0.953213, 0.777482}
```

```
f[0.3]
```

```
{0.299758, 0.00897463}
```

```
ParametricPlot[f[t], {t, -10, 10}]
```

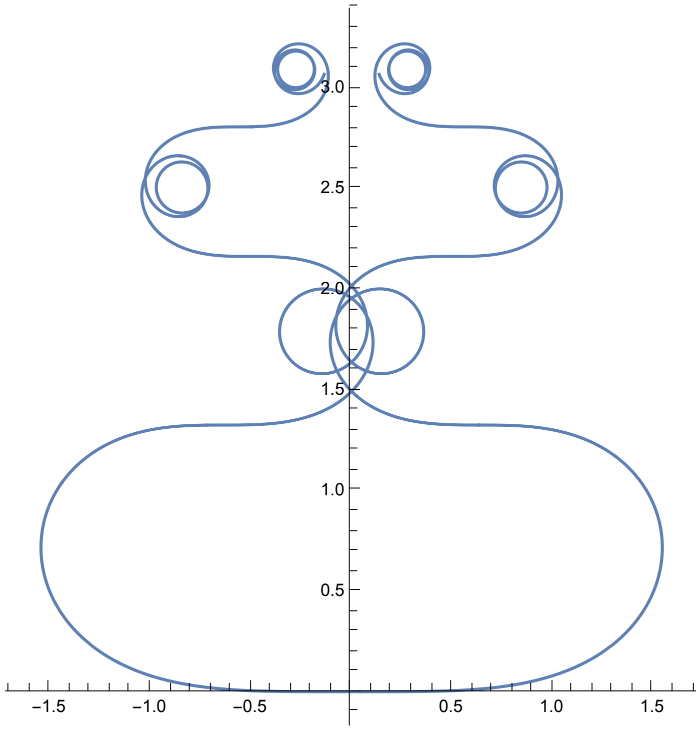


Here is a curve whose curvature is s sin(s).

```
g[t_] = Through[intrinsic[(# Sin[#]) &, 0, {0, 0, 0}, {-12, 12}][t]]
```

```
{InterpolatingFunction[ Domain: {{-12., 12.}} Output: scalar ][t],
 InterpolatingFunction[ Domain: {{-12., 12.}} Output: scalar ][t]}
```

```
ParametricPlot[g[t], {t, -12, 12}, PlotPoints → 100]
```



Drawing curves in space with Assigned Curvature and Torsion

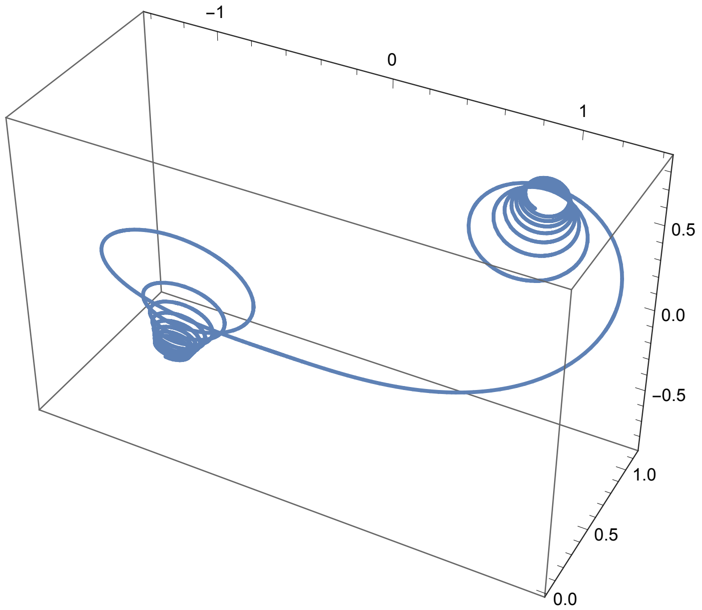
Fundamental Theorem of Plane Curves

Let α and γ be unit curves in $\mathbb{R}^3$ definded on the same interval (a, b) and suppose α and β they have the same torsion and positive curvature. Then there is an Euclidean motion mapping α into γ. Given piecwise - continuous function $\kappa : (a, b) \to \mathbb{R}$ and $t : (a, b) \to \mathbb{R}$ be differentiable functions with $\kappa > 0$. Then there exists a unit speed curve whose curvature and torsion are κ and t. For $a < s_0 < b$ the value $\beta(s_0)$ can be prescribed arbitrarily. Also, the values $T(s_0)$ and $N(s_0)$ can be prescribed subject to the conditions that $T(s_0) = N(s_0) = 1$ and $T(s_0).N(s_0) = 0$.

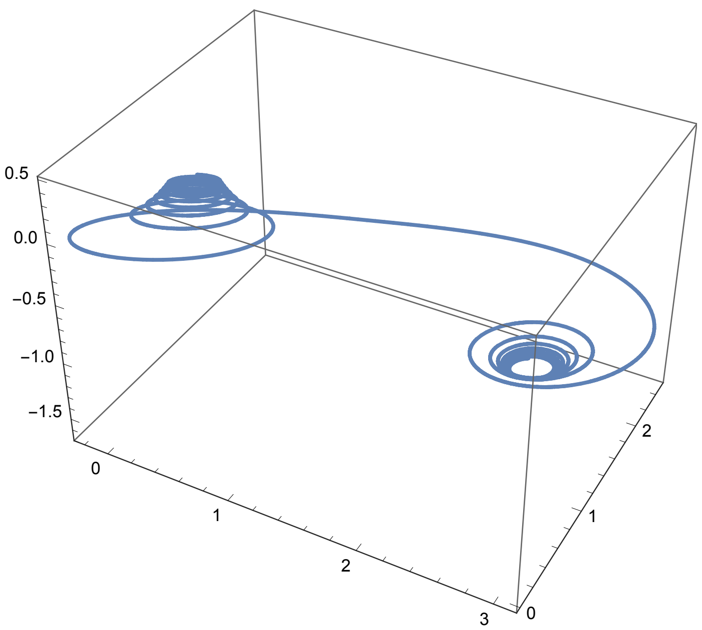
```
Clear[plotintrinsic3d]
```

```
plotintrinsic3d[{kk_, tt_},
  {a_: 0, {p1_: 0, p2_: 0, p3_: 0}, {q1_: 1, q2_: 0, q3_: 0}, {r1_: 0, r2 : 1, r3_: 0}},
  {smin_: - 10, smax_: 10}, opts___] :=
 ParametricPlot3D[Evaluate[Module[{x1, x2, x3, t1, t2, t3, n1, n2, n3, b1, b2, b3},
    {x1[s], x2[s], x3[s]} /. NDSolve[{x1'[ss] == t1[ss], x2'[ss] == t2[ss],
       x3'[ss] == t3[ss], t1'[ss] == kk[ss] n1[ss], t2'[ss] == kk[ss] n2[ss],
       t3'[ss] == kk[ss] n3[ss], n1'[ss] == -kk[ss] t1[ss] + tt[ss] b1[ss],
       n2'[ss] == -kk[ss] t2[ss] + tt[ss] b2[ss],
       n3'[ss] == -kk[ss] t3[ss] + tt[ss] b3[ss], b1'[ss] == -tt[ss] n1[ss],
       b2'[ss] == -tt[ss] n2[ss], b3'[ss] == -tt[ss] n3[ss], x1[a] == p1, x2[a] == p2,
       x3[a] == p3, t1[a] == q1, t2[a] == q2, t3[a] == q3, n1[a] == r1, n2[a] == r2,
       n3[a] == r3, b1[a] == q2 r3 - q3 r2, b2[a] == q3 r1 - q1 r3, b3[a] == q1 r2 - q2 r1},
      {x1, x2, x3, t1, t2, t3, n1, n2, n3, b1, b2, b3},
      {ss, smin, smax}]]], {s, smin, smax}, opts]
```

```
plotintrinsic3d[{Abs[#] &, 0.3 &}, {{0, 0}, {1, 0}, {0, 1}}, {}, PlotPoints → 500]
```




```
plotintrinsic3d[{Abs[#] &, 0.3 &}, {2, {0, 0}, {1, 0}, {1, 1}}, {}, PlotPoints → 500]
```



```
plotintrinsic3d[{1.3 &, Sin[#] &},
 {0, {0, 0, 0}, {1, 0, 0}, {0, 1, 0}}, {-10 Pi, 10 Pi}, PlotPoints → 500]
```