Start

Vi starter på nytt

```
restart :
```

Vi lader inn kommandopakken

```
with(Student[VectorCalculus])
```

[&x, `*`, `+`, `-`, `.`, <,>, <|>, About, ArcLength, BasisFormat, Binormal, ConvertVector, CrossProduct, Curl, Curvature, D, Del, DirectionalDiff, Divergence, DotProduct, FlowLine, Flux, GetCoordinates, GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, Jacobian, Laplacian, LineInt, MapToBasis, Nabla, Norm, Normalize, PathInt, PlotPositionVector, PlotVector, PositionVector, PrincipalNormal, RadiusOfCurvature, RootedVector, ScalarPotential, SetCoordinates, SpaceCurve, SpaceCurveTutor, SurfaceInt, TNBFrame, Tangent, TangentLine, TangentPlane, TangentVector, Torsion, Vector, VectorField, VectorFieldTutor, VectorPotential, VectorSpace, diff, evalVF, int, limit, series] **(1.1)**

Vi lader inn kommandopakken

```
with(plots)
```

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot] **(1.2)**

Bevegelse i polarkoordinater

Vi lager en MAPLE-funksjon som tegner kurven for a <= t <= b og når t = i tegner den u_r og u_θ også

```
BevegelsePolar := proc(r, θ, a, b, i)

# lokale variabler
 local P0, P1, P2, P3, T, N, ν, κ;

# a er startverdien til t
 # b er sluttverdien til t
 # i er t verdien når vi tegner vektorene
```

```
# kurven
P0 := SpaceCurve(⟨r(t)·cos(θ(t)), r(t)·sin(θ(t))⟩, t = a ..b, axes = normal, labels = ['x','y'],
    numpoints = 1000);

# posisjonsvektoren
P1 := arrow(⟨0, 0⟩, ⟨r(i)·cos(θ(i)), r(i)·sin(θ(i))⟩, shape = arrow, color ='black') ;

# u_r
P2 := arrow(⟨r(i)·cos(θ(i)), r(i)·sin(θ(i))⟩, ⟨cos(θ(i)), sin(θ(i))⟩, shape = arrow, color =
    'blue', length = 1) :

# u_θ
P3 := arrow(⟨r(i)·cos(θ(i)), r(i)·sin(θ(i))⟩, ⟨-sin(θ(i)), cos(θ(i))⟩, shape = arrow, color =
    'red', length = 1) :

# alle sammen
display(P0, P1, P2, P3, scaling = constrained);
end;
```

```
proc(r, theta, a, b, i)                                                                    (2.1)
    local P0, P1, P2, P3, T, N, nu, kappa;
    P0 := `Student:-VectorCalculus`:-SpaceCurve(`Student:-VectorCalculus`:-`<,>`(r(t)
     * cos(theta(t)), r(t) * sin(theta(t))), t = a ..b, axes = normal, labels = ['x', 'y'], numpoints
     = 1000);
    P1 := plots:-arrow(`Student:-VectorCalculus`:-`<,>`(0, 0), `Student:-VectorCalculus`:-
    `<,>`(r(i) * cos(theta(i)), r(i) * sin(theta(i))), shape = plots:-arrow, color = 'black');
    P2 := plots:-arrow(`Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i)), r(i)
     * sin(theta(i))), `Student:-VectorCalculus`:-`<,>`(cos(theta(i)), sin(theta(i))), shape
     = plots:-arrow, color = 'blue', length = 1);
    P3 := plots:-arrow(`Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i)), r(i)
     * sin(theta(i))), `Student:-VectorCalculus`:-`<,>`(`Student:-VectorCalculus`:-
    `-`(sin(theta(i))), cos(theta(i))), shape = plots:-arrow, color = 'red', length = 1);
    plots:-display(P0, P1, P2, P3, scaling = constrained)
end proc
```

Vi skal jobbe med den følgende parameterfremstillingen

$r := t \to 4 \cdot \exp(\cos(t))$

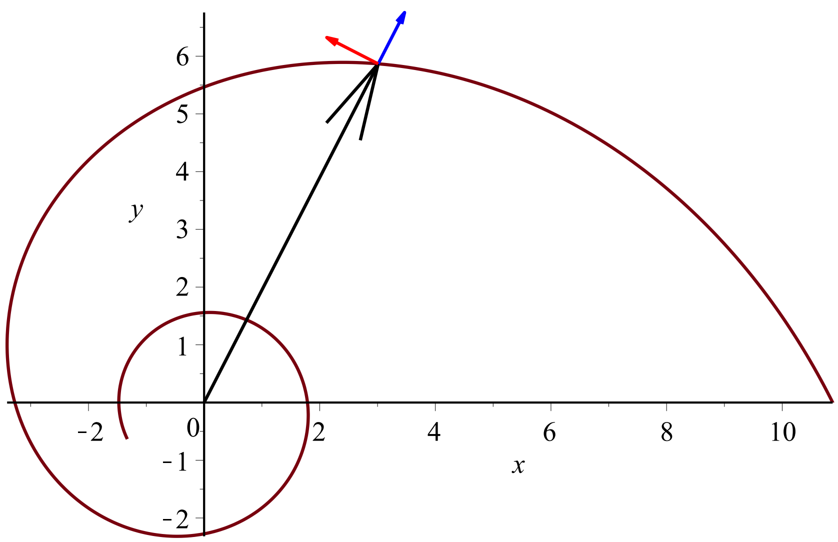
$$t \to 4\, e^{\cos(t)} \tag{2.2}$$

$\theta := t \to t^2$

$$t \to t^2 \tag{2.3}$$

Vi får det følgende bildet

$BevegelsePolar\left(r, \theta, 0, \pi, \pi \frac{1}{3}\right)$

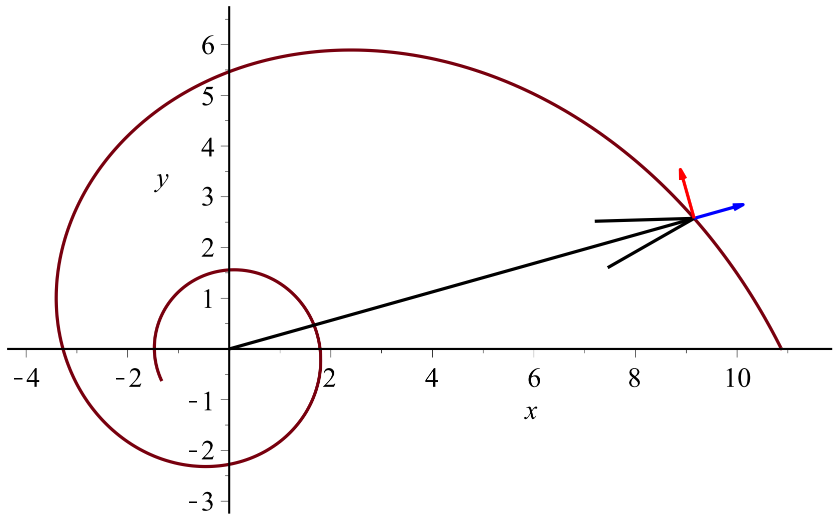


Graph of the curve represented parametrically by the components of the given vector.

Vi lager en animasjon

***animate*(*BevegelsePolar*, [(*r*, θ, 0, Pi, *n*)], *n* = 0 ..Pi, *scaling* = *constrained*)**

$n = 0.52360$



Graph of the curve represented parametrically by the components of the given vector.

Bevegelse i sylinderkoordinater

Vi lager en MAPLE-funksjon som tegner kurven for a <= t <= b og når t = i tegner den u_r, u_θ og k også sammen med u_r u_θ planet

```
BevegelseSylinder := proc(r, θ, z, a, b, i)

# lokale variabler
 local P0, P1, P2, P3, P4;

# a er startverdien til t
 # b er sluttverdien til t
 # i er t verdien når vi tegner vektorene

 # kurven
P0 := SpaceCurve(⟨r(t)·cos(θ(t)), r(t)·sin(θ(t)), z(t)⟩, t = a..b, axes = normal, labels = ['x','y
      ','z'], numpoints = 1000);
```

```
# posisjonsvektoren
P1 ≔ arrow( ⟨0, 0, 0⟩, ⟨r(i) · cos( θ(i) ), r(i) · sin( θ(i) ), z(i)⟩, shape = arrow, color ='black' ) ;

# u_r
P2 ≔ arrow( ⟨r(i) · cos( θ(i) ), r(i) · sin( θ(i) ), z(i)⟩, ⟨cos( θ(i) ), sin( θ(i) ), 0⟩, shape = arrow,
    color ='blue', length = 1 ) :

# u_θ
P3 ≔ arrow( ⟨r(i) · cos( θ(i) ), r(i) · sin( θ(i) ), z(i)⟩, ⟨ - sin( θ(i) ), cos( θ(i) ), 0⟩, shape
    = arrow, color ='red', length = 1 ) :

# k
P4 ≔ arrow( ⟨r(i) · cos( θ(i) ), r(i) · sin( θ(i) ), z(i)⟩, ⟨0, 0, 1⟩, shape = arrow, color ='green',
    length = 1 ) :

# u_r og u_θ planet
P5 ≔ implicitplot3d(z = z(i), x = r(i) · cos( θ(i) ) − 1.5 ..r(i) · cos( θ(i) ) + 1.5, y = r(i) · sin( θ(i) )
    − 1.5 ..r(i) · sin( θ(i) ) + 1.5, z = z(i) − 0.1 .. z(i) + 0.1, color = cyan, transparency = 0.95 ) :

# alle sammen
display(P0, P1, P2, P3, P4, P5, scaling = constrained);
end;
Warning, `P5` is implicitly declared local to procedure
`BevegelseSylinder`
proc(r, theta, z, a, b, i)                                                                    (3.1)
    local P0, P1, P2, P3, P4, P5;
    P0 := `Student:-VectorCalculus`:-SpaceCurve( `Student:-VectorCalculus`:-`<,>`(r(t)
    * cos(theta(t) ), r(t) * sin(theta(t) ), z(t) ), t = a ..b, axes = normal, labels = ['x', 'y', 'z'],
    numpoints = 1000 );
    P1 := plots:-arrow( `Student:-VectorCalculus`:-`<,>`(0, 0, 0 ),
    `Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i) ), r(i) * sin(theta(i) ), z(i) ), shape
    = plots:-arrow, color = 'black' );
    P2 := plots:-arrow( `Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i) ), r(i)
    * sin(theta(i) ), z(i) ), `Student:-VectorCalculus`:-`<,>`(cos(theta(i) ), sin(theta(i) ),
    0 ), shape = plots:-arrow, color = 'blue', length = 1 );
    P3 := plots:-arrow( `Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i) ), r(i)
    * sin(theta(i) ), z(i) ), `Student:-VectorCalculus`:-`<,>`( `Student:-VectorCalculus`:-
    `-`(sin(theta(i) ) ), cos(theta(i) ), 0 ), shape = plots:-arrow, color = 'red', length = 1 );
    P4 := plots:-arrow( `Student:-VectorCalculus`:-`<,>`(r(i) * cos(theta(i) ), r(i)
    * sin(theta(i) ), z(i) ), `Student:-VectorCalculus`:-`<,>`(0, 0, 1 ), shape = plots:-arrow,
    color = 'green', length = 1 );
    P5 := plots:-implicitplot3d(z = z(i), x = r(i) * cos(theta(i) ) − 1.5 ..r(i) * cos(theta(i) )
    + 1.5, y = r(i) * sin(theta(i) ) − 1.5 ..r(i) * sin(theta(i) ) + 1.5, z = z(i) − 0.1 ..z(i)
```

```
+ 0.1, color = cyan, transparency = 0.95 );
plots:-display(P0, P1, P2, P3, P4, P5, scaling = constrained)
end proc
```

Vi skal jobbe med den følgende parameterfremstillingen

$r := t \rightarrow 4 \cdot \exp(\cos(t))$

$$t \rightarrow 4\, e^{\cos(t)} \tag{3.2}$$

$\theta := t \rightarrow t^2$

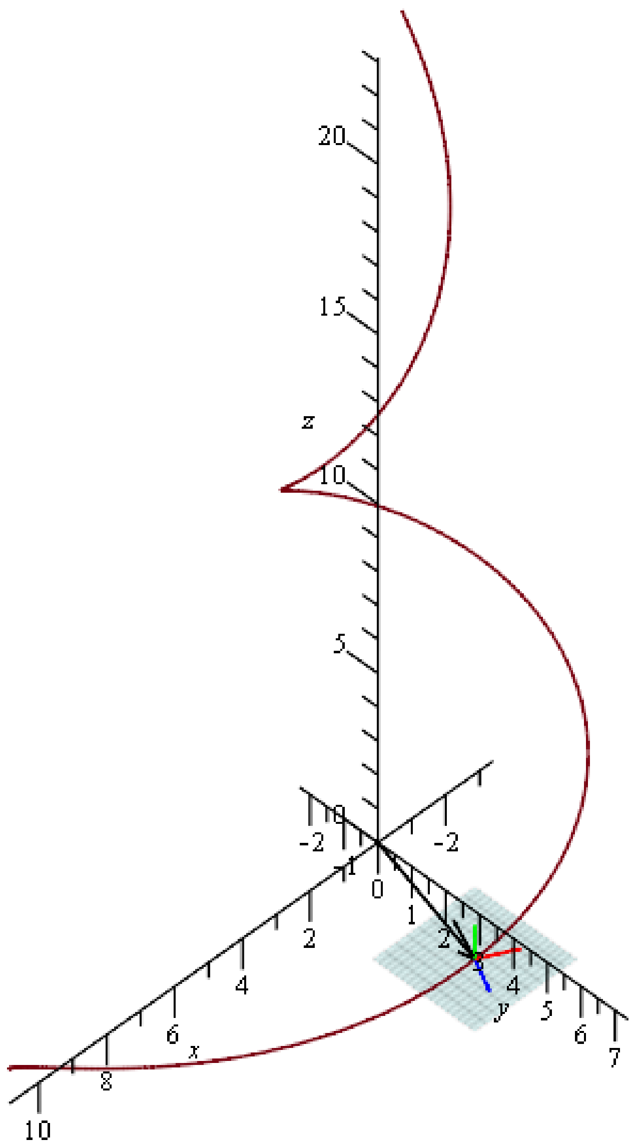
$$t \rightarrow t^2 \tag{3.3}$$

$z := t \rightarrow \exp(t)$

$$t \rightarrow e^t \tag{3.4}$$

Vi får det følgende bildet

$BevegelseSylinder\left(r, \theta, z, 0, \pi, \pi \frac{1}{3}\right)$



z
20
15
10
5
0
-2
-1
0
1
2
4
5
6
7
y
-2
2
4
6
8
10
x

Vi lager en animasjon

$animate(BevegelseSylinder, [(r, \theta, z, 0, \mathrm{Pi}, n)], n = 0..\mathrm{Pi}, scaling = constrained)$

$n = 1.8326$