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)**

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*] **(2)**

Vi lager en vektorfunksjon som beskriver en kurve (sirkel rundt <0,0,1> på planen z=1 med radius 2)

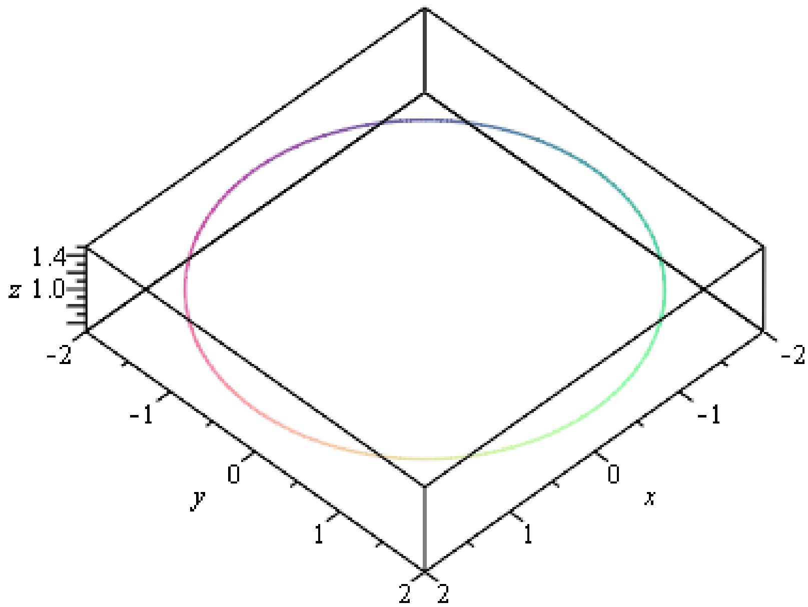
$r := t \rightarrow \langle 2\cos(t), 2\sin(t), 1\rangle$

$$t \rightarrow VectorCalculus{:-}{<,>}(2\cos(t), 2\sin(t), 1) \quad \textbf{(3)}$$

Vi kan tegne kurven når t går fra 0 til 2Pi

- vi setter skala fast (scaling = constrained)
- vi vil gjerne se på aksene som en boks (axes = boxed)
- aksene heter x, y og z-aksen (labels=['x','y','z']
- vi tegner 1000 punkter på kurven (numpoints = 1000)

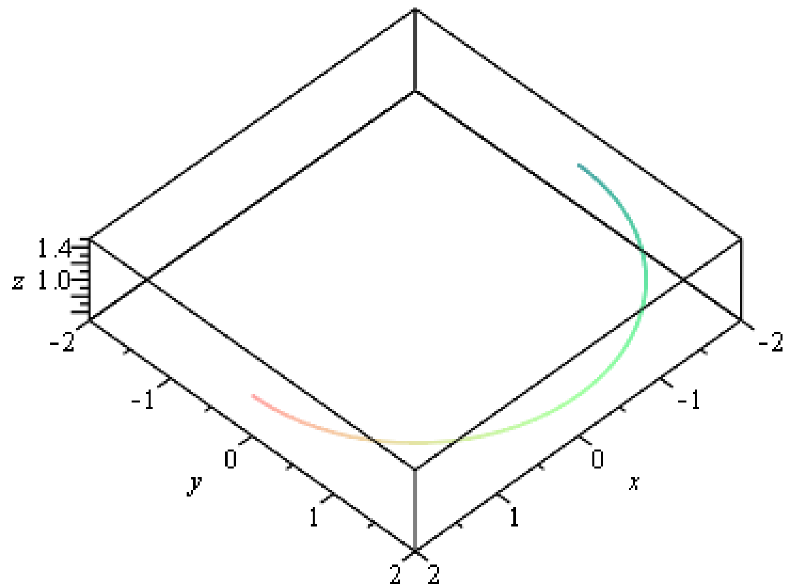
spacecurve(*r*(*t*), *t* = 0 ..2 Pi, *scaling* = *constrained*, *axes* = *boxed*, *labels* = ['*x*','*y*','*z*'], *numpoints* = 1000)



Vi vil gjerne se hvordan vi går langs kurven -> Vi lager en animasjon

$animate(spacecurve, [(r(t), t = 0..n, scaling = constrained, axes = boxed, labels = ['x','y','z'], numpoints = 1000)], n = 0..2\,\mathrm{Pi})$

$n = 3.1416$



Vi får enhetstangenten med kommando TangentVector (enhetsvektor <-> normalized = true)

$\boldsymbol{T := TangentVector(r, normalized = true);}$

$$t \rightarrow VectorCalculus{:-}VectorSpace(cartesian, [2\cos(t), 2\sin(t), 1]){:-}Vector([-\sin(t), \cos(t), 0]) \tag{4}$$

$\boldsymbol{T(0)}$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{5}$$

$\boldsymbol{T(1)}$

$$\begin{bmatrix} -\sin(1) \\ \cos(1) \\ 0 \end{bmatrix} \tag{6}$$

Det er en enhetsvektor, siden normen sin er 1

$\boldsymbol{Norm(T(0))}$

$$1 \tag{7}$$

$\boldsymbol{Norm(T(1))}$

$$1 \quad (8)$$

Når vi jobber uten datamaskin, vi får enhetstangenten fra formelen: T = v / |v|

- v er hastighetsvektoren
- |v| er farten

Vi får hastigheten når vi deriverer r

$diff(r(t), t)$

$$-2\sin(t)e_x + 2\cos(t)e_y \quad (9)$$

Farten er hastighetens størrelse (hastighetens norme)

$Norm(diff(r(t), t))$

$$2 \quad (10)$$

Vi får enhetstangenten når vi deler hastigheten med farten

$\frac{diff(r(t), t)}{Norm(diff(r(t), t))}$

$$-\sin(t)e_x + (\cos(t))e_y \quad (11)$$

Vi får krumningen med kommando Curvature

- når vi jobber uten datamaskin, da er det 1/|v| |dT/dt|

$\kappa = Curvature(r)$

$$\kappa = \left(t \rightarrow \frac{1}{2}\sqrt{\sin(t)^2 + \cos(t)^2}\right) \quad (12)$$

Vi får normalvektoren med kommando PrincipalNormal

- når vi jobber uten datamaskin, da er det 1/κ dT/ds eller dT/dt / |dT/dt|

$N := PrincipalNormal(r, normalized = true);$

$t \rightarrow VectorCalculus{:-}VectorSpace(cartesian, [2\cos(t), 2\sin(t), 1]){:-}Vector([-\cos(t), -\sin(t), 0])$ (13)

$N(1)$

$$\begin{bmatrix} -\cos(1) \\ -\sin(1) \\ 0 \end{bmatrix} \quad (14)$$

Det er en enhetsvektor, siden normen sin er 1

$Norm(N(1))$

$$1 \quad (15)$$

Vi lager et bilde fra kurven ved 0 <= t <= Pi

$P1 := spacecurve(r(t), t = 0..Pi, axes = boxed, labels = ['x','y','z'], numpoints = 1000);$

$$PLOT3D(...) \quad (16)$$

Vi lager et bilde av enhetstangenten ved t = Pi/2 (fargen blir blå)

$T := TangentVector(r, normalized = true);$

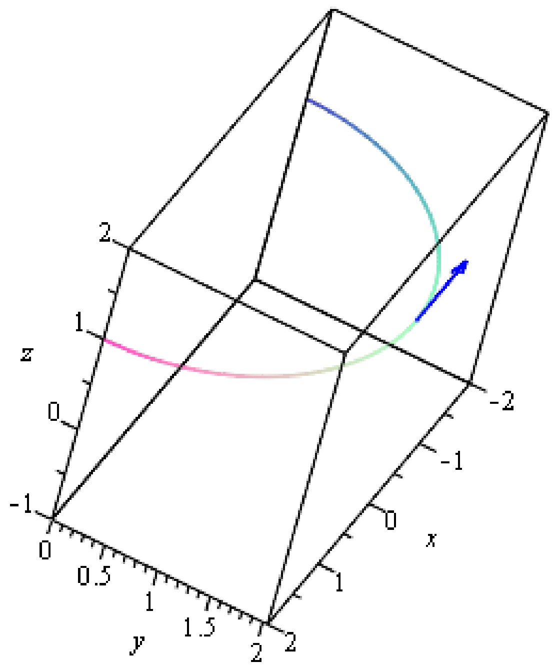
$P2 := arrow\left(r\left(\frac{Pi}{2}\right), T\left(\frac{Pi}{2}\right), shape = arrow, color = 'blue'\right);$

$t \rightarrow VectorCalculus{:-}VectorSpace(cartesian, [2\cos(t), 2\sin(t), 1]){:-}Vector([-\sin(t), \cos(t), 0])$

$$PLOT3D(...) \quad (17)$$

Vi viser dem sammen

display(*P1*, *P2*, *scaling* = *constrained*)



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

BevegelseT ≔ **proc**(*r*, *a*, *b*, *i*)

 # lokale variabler

local *P0*, *P1*, *P2*, *T*, ν;

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

 # kurven
P0 ≔ *spacecurve*(*r*(*t*), *t* = *a* ..*b*, *axes* = *boxed*, *labels* = ['x','y','z'], *numpoints* = 1000) :

 # posisjonsvektoren
P1 ≔ *arrow*(⟨0, 0, 0⟩, *r*(*i*), *shape* = *arrow*, *color* ='*black*') ;

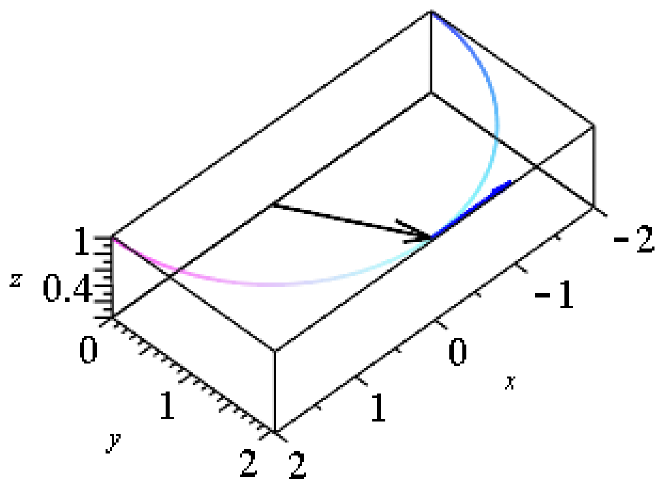
 # farten
ν ≔ *diff* (*r*(**τ**), **τ**);

```
# enhetstangenten
T ≔ TangentVector(r, normalized = true);
P2 ≔ arrow(r(i), T(i), shape = arrow, color ='blue') :

# alle sammen
display(P0, P1, P2, scaling = constrained, caption = typeset("v=%1", Norm(subs(τ = i, ν))), font
    = [TIMES, 16]);
end;
```

```
proc(r, a, b, i)                                                                 (18)
    local P0, P1, P2, T, nu;
    P0 := plots:-spacecurve(r(t), t = a ..b, axes = boxed, labels = ['x', 'y', 'z'], numpoints = 1000);
    P1 := plots:-arrow(VectorCalculus:-`<,>`(0, 0, 0), r(i), shape = plots:-arrow, color = 'black
    ');
    nu := diff(r(tau), tau);
    T := VectorCalculus:-TangentVector(r, normalized = true);
    P2 := plots:-arrow(r(i), T(i), shape = plots:-arrow, color = 'blue');
    plots:-display(P0, P1, P2, scaling = constrained, caption = typeset("v=%1", VectorCalculus:-
    Norm(subs(tau = i, nu))), font = [TIMES, 16])
end proc
```

Vi får et liknende bilde ved BevegelseT

$BevegelseT\left(r, 0, \mathrm{Pi}, \frac{\mathrm{Pi}}{2}\right)$

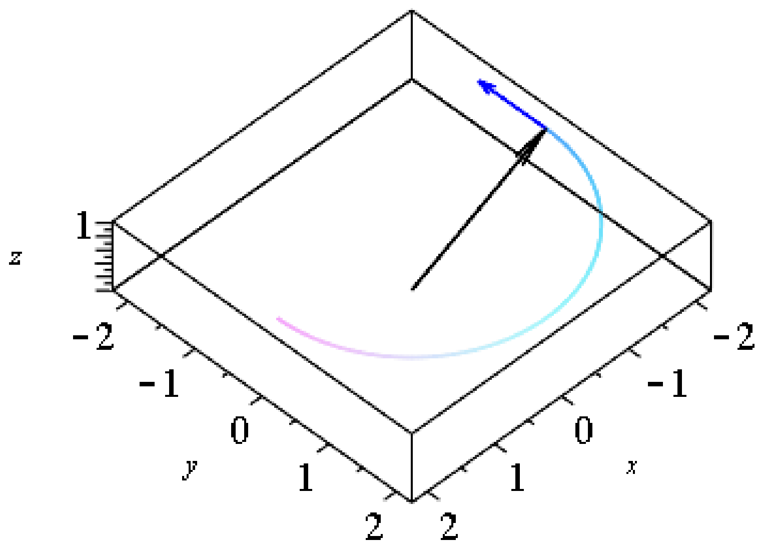


v=2

Vi kan lage en animasjon når vi tegner kurven sammen med enhetstangenten

$animate\left(BevegelseT, \left[\left(r, 0, n, n \right) \right], n = 0 ..2\,\pi, font = \left[TIMES, 16 \right] \right)$

$$n = 3.1416$$

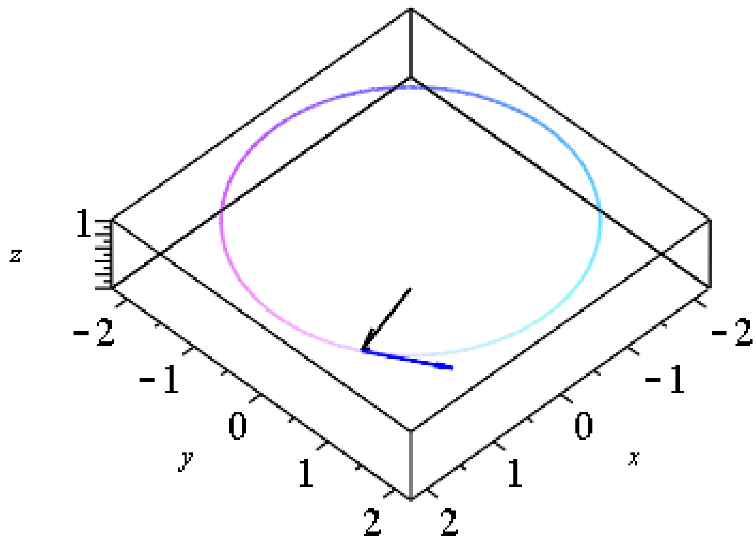


v=2.

Vi kan lage en animasjon når vi tegner enhetstangenten langs kurven

animate(*BevegelseT*, [(*r*, 0, 2 Pi, *n*)], *n* = 0 ..2 π, *font* = [*TIMES*, 16])

$$n = 0.52360$$



v=2.

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

```
BevegelseTN := proc(r, a, b, i)

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

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

  # kurven
  P0 := spacecurve(r(t), t = a ..b, axes = boxed, labels = [ 'x','y','z' ], numpoints = 1000 ) :

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

 # farten
 v := diff (r(τ), τ);

# enhetstangenten
```

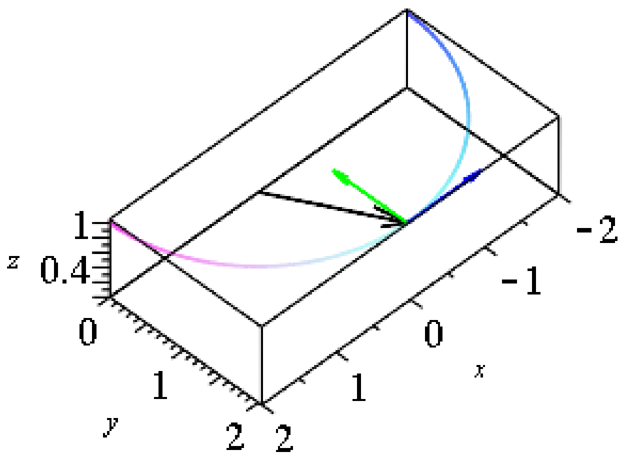
```
T := TangentVector(r, normalized = true) :
P2 := arrow(r(i), T(i), shape = arrow, color ='blue') :

# krumningen
κ := Curvature(r) :

# normalvektoren
N := PrincipalNormal(r, normalized = true) :
P3 := arrow(r(i), N(i), shape = arrow, color ='green') :

# alle sammen
display(P0, P1, P2, P3, scaling = constrained, caption = typeset("v=%1, κ=%2", Norm(subs(τ = i,
    ν)), κ(i)), font = [TIMES, 16]);
end;
```

proc(r, a, b, i) **(19)**

```
    local P0, P1, P2, P3, T, N, nu, kappa;
    P0 := plots:-spacecurve(r(t), t = a ..b, axes = boxed, labels = ['x', 'y', 'z'], numpoints = 1000);
    P1 := plots:-arrow(VectorCalculus:-`<,>`(0, 0, 0), r(i), shape = plots:-arrow, color = 'black
    ');
    nu := diff(r(tau), tau);
    T := VectorCalculus:-TangentVector(r, normalized = true);
    P2 := plots:-arrow(r(i), T(i), shape = plots:-arrow, color = 'blue');
    kappa := VectorCalculus:-Curvature(r);
    N := VectorCalculus:-PrincipalNormal(r, normalized = true);
    P3 := plots:-arrow(r(i), N(i), shape = plots:-arrow, color = 'green');
    plots:-display(P0, P1, P2, P3, scaling = constrained, caption = typeset("v=%1, κ=%2",
    VectorCalculus:-Norm(subs(tau = i, nu)), kappa(i)), font = [TIMES, 16])
end proc
```

BevegelseTN gir dette bildet

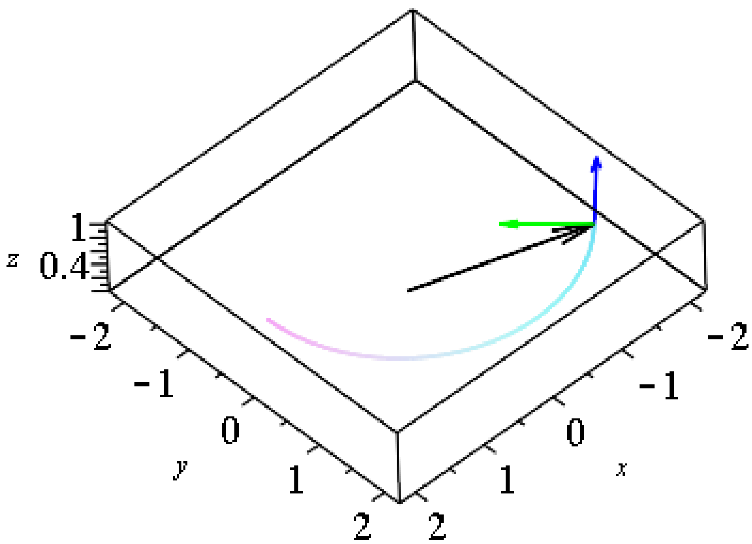
$$BevegelseTN\left(r, 0, \mathrm{Pi}, \frac{\mathrm{Pi}}{2}\right)$$



$$\text{v}=2,\quad = \frac{1}{2}$$

Vi kan lage en animasjon når vi tegner kurven sammen med enhetstangenten og normalvetkoren
animate(***BevegelseTN*, [(*r*, 0, *n*, *n*)], *n* = 0 ..2 Pi, *font* = [*TIMES*, 16]**)

$$n = 2.3562$$

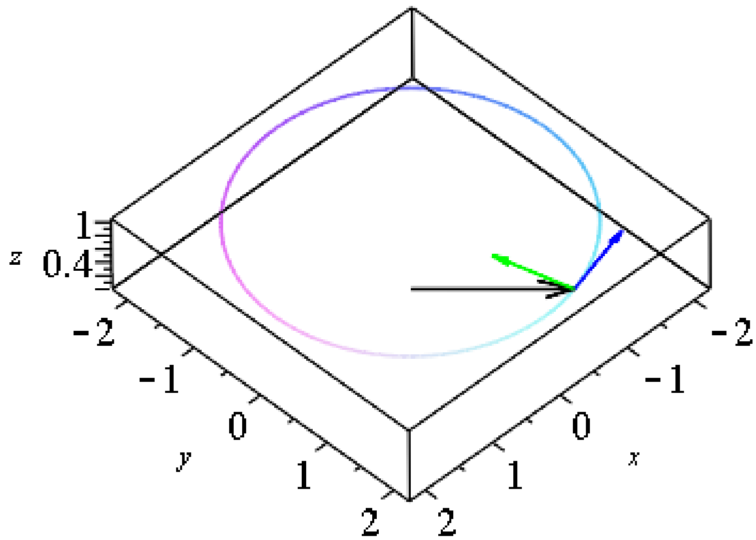


$$\text{v=2.,} \quad = 0.5000000000$$

Vi kan lage en animasjon når vi tegner enhetstangenten og normalvektoren langs kurven

$animate\left(BevegelseTN, [(r, 0, 2\,\mathrm{Pi}, n)], n = 0 ..2\,\pi, font = [TIMES, 16]\right)$

$n = 1.8326$



v=2., $= 0.5000000000$

Vi skal jobbe med heliksen

$r2 := t \to \langle \cos(t), \sin(t), t \rangle$

$$t \to VectorCalculus:- <,>(\cos(t), \sin(t), t) \qquad (20)$$

Vi lager en MAPLE-funksjon som tegner kurven for a <= t <= b og når t = i tegner den enhetstangenten, normalvektoren og krumningssirkelen også

```
BevegelseTNS := proc(r, a, b, i)

# lokale variabler
 local P0, P1, P2, P3, P4, P5, T, N, v, κ, R;

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

  # kurven
 P0 := spacecurve(r(t), t = a ..b, axes = boxed, labels = [ 'x','y','z' ], numpoints = 1000 ) :

 # posisjonsvektoren
 P1 := arrow( ⟨0, 0, 0⟩ , r(i), shape = arrow, color ='black' ) ;
```

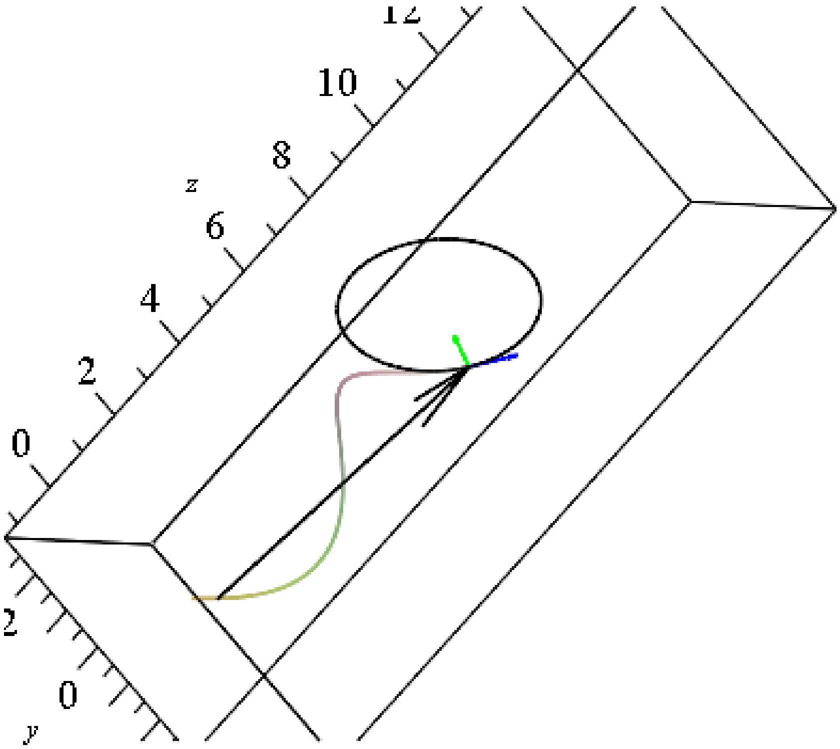
```
# farten
ν ≔ diff(r(η), η);

# enhetstangenten
T ≔ TangentVector(r, normalized = true) :
P2 ≔ arrow(r(i), T(i), shape = arrow, color ='blue') :

# krumningen
κ ≔ Curvature(r) :

# normalvektoren
N ≔ PrincipalNormal(r, normalized = true) :
P3 ≔ arrow(r(i), N(i), shape = arrow, color ='green') :

# krumningssirkelen
R ≔ RadiusOfCurvature(r) :
P4 ≔ spacecurve(r(i) + R(i)·N(i) + R(i)·cos(τ)·T(i) + R(i)·sin(τ)·N(i), τ = 0 ..2 Pi, color =
    'black') :

# alle sammen
display(P0, P1, P2, P3, P4, scaling = constrained, caption = typeset("ν=%1, κ=%2", Norm(subs(η
    = i, ν)), R(i)), font = [TIMES, 16]);
end;
```

```
proc(r, a, b, i)                                                                    (21)
    local P0, P1, P2, P3, P4, P5, T, N, nu, kappa, R;
    P0 := plots:-spacecurve(r(t), t = a ..b, axes = boxed, labels = ['x', 'y', 'z'], numpoints = 1000);
    P1 := plots:-arrow(VectorCalculus:-`<,>`(0, 0, 0), r(i), shape = plots:-arrow, color = 'black
    ');
    nu := diff(r(eta), eta);
    T := VectorCalculus:-TangentVector(r, normalized = true);
    P2 := plots:-arrow(r(i), T(i), shape = plots:-arrow, color = 'blue');
    kappa := VectorCalculus:-Curvature(r);
    N := VectorCalculus:-PrincipalNormal(r, normalized = true);
    P3 := plots:-arrow(r(i), N(i), shape = plots:-arrow, color = 'green');
    R := VectorCalculus:-RadiusOfCurvature(r);
    P4 := plots:-spacecurve(r(i) + R(i) * N(i) + R(i) * cos(tau) * T(i) + R(i) * sin(tau)
     * N(i), tau = 0 ..2 * Pi, color = 'black');
    plots:-display(P0, P1, P2, P3, P4, scaling = constrained, caption = typeset("ν=%1, κ=%2",
    VectorCalculus:-Norm(subs(eta = i, nu)), R(i)), font = [TIMES, 16])
end proc
```

Vi kan lage en animasjon når vi tegner kurven sammen med enhetstangenten, normalvetkorenog krumningssirkelen

```
animate(BevegelseTNS, [(r2, 0, n , n)], n = 0 ..4 Pi, font = [TIMES, 16])
```

$$n = 7.3304$$

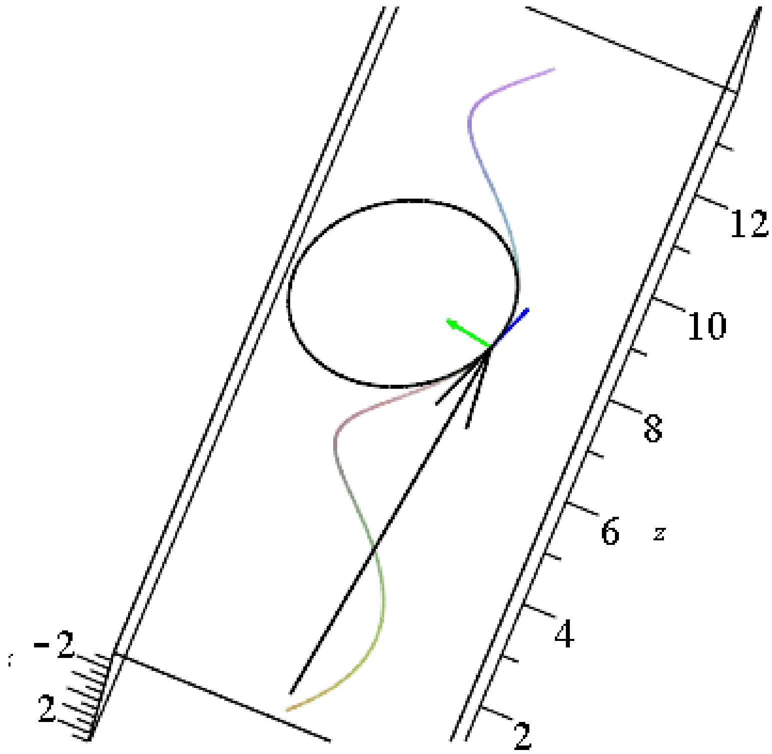


v=1.414213562,   =2

Vi kan lage en animasjon når vi tegner enhetstangenten, normalvektoren og krumningssirkelen langs kurven

$animate(BevegelseTNS, [(r2, 0, 4\ \mathrm{Pi}, n)], n = 0 ..4\ \mathrm{Pi}, font = [TIMES, 16])$

$n = 7.3304$



v=1.414213562, =2

Vi lager en MAPLE-funksjon som tegner kurven for a <= t <= b og når t = i tegner den enhetstangenten, normalvektoren og binormalen også

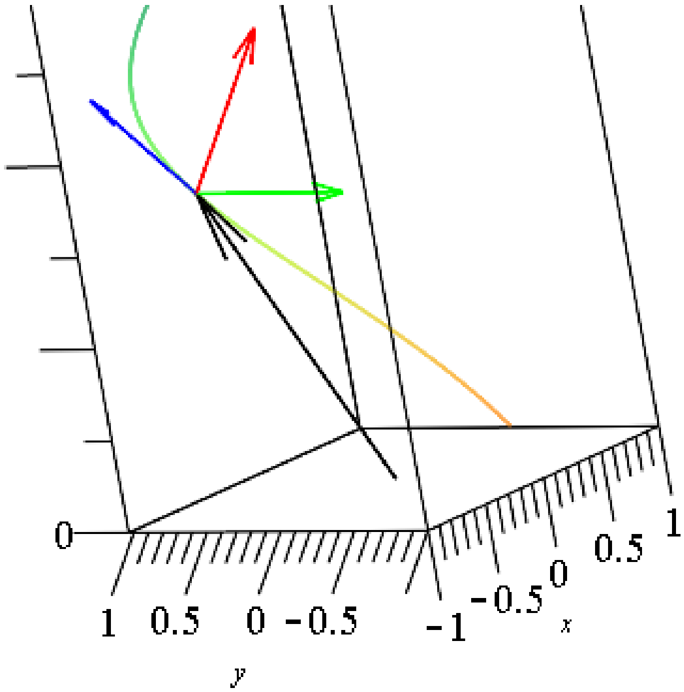
```
BevegelseTNB := proc(r, a, b, i)

 # lokale variabler
 local P0, P1, P2, P3, P4, T, N, B, v, κ, τ;

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

 # kurven
 P0 := spacecurve(r(t), t = a ..b, axes = boxed, labels = [ 'x','y','z' ], numpoints = 5000) :

 # posisjonsvektoren
 P1 := arrow(⟨0, 0, 0⟩, r(i), shape = arrow, color ='black') :

 # farten
 v := diff(r(η), η);
```

```
# enhetstangenten
T := TangentVector(r, normalized = true) :
P2 := arrow(r(i), T(i), shape = arrow, color ='blue') :

# krumningen
κ := Curvature(r) :

# normalvektoren
N := PrincipalNormal(r, normalized = true) :
P3 := arrow(r(i), N(i), shape = arrow, color ='green') :

# torsjonen
τ := Torsion(r) :

# binormalen
B := Binormal(r, normalized = true) :
P4 := arrow(r(i), B(i), shape = arrow, color ='red') :

# alle sammen
display(P0, P1, P2, P3, P4, scaling = constrained, caption = typeset("ν = %1, κ = %2, τ = %3",
    Norm(subs(η = i, ν)), κ(i), τ(i)), font = [TIMES, 16]);
end;
```

```
proc(r, a, b, i)                                                                    (22)
    local P0, P1, P2, P3, P4, T, N, B, nu, kappa, tau;
    P0 := plots:-spacecurve(r(t), t = a ..b, axes = boxed, labels = ['x', 'y', 'z'], numpoints = 5000);
    P1 := plots:-arrow(VectorCalculus:-`<,>`(0, 0, 0), r(i), shape = plots:-arrow, color = 'black
    ');
    nu := diff(r(eta), eta);
    T := VectorCalculus:-TangentVector(r, normalized = true);
    P2 := plots:-arrow(r(i), T(i), shape = plots:-arrow, color = 'blue');
    kappa := VectorCalculus:-Curvature(r);
    N := VectorCalculus:-PrincipalNormal(r, normalized = true);
    P3 := plots:-arrow(r(i), N(i), shape = plots:-arrow, color = 'green');
    tau := VectorCalculus:-Torsion(r);
    B := VectorCalculus:-Binormal(r, normalized = true);
    P4 := plots:-arrow(r(i), B(i), shape = plots:-arrow, color = 'red');
    plots:-display(P0, P1, P2, P3, P4, scaling = constrained, caption
     = typeset("ν = %1, κ = %2, τ = %3", VectorCalculus:-Norm(subs(eta = i, nu)), kappa(i),
    tau(i)), font = [TIMES, 16])
end proc
```

Vi skal jobbe med heliksen

$BevegelseTNB\left(r2, 0, 3\,\mathrm{Pi}, \frac{\mathrm{Pi}}{2}\right)$

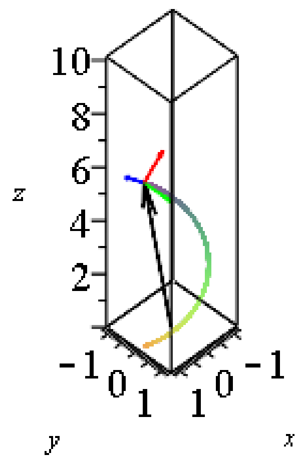


$$\mathrm{v} = \sqrt{2}, \quad = \frac{1}{2}, \quad = \frac{1}{2}$$

Vi kan lage en animasjon når vi tegner kurven sammen med enhetstangenten, normalvetkoren og binormalen

$\color{red}{\boldsymbol{animate\left(BevegelseTNB, [(r2, 0, n, n)], n = 0 ..3\,\pi, font = [TIMES, 16]\right)}}$

$n = 4.7124$

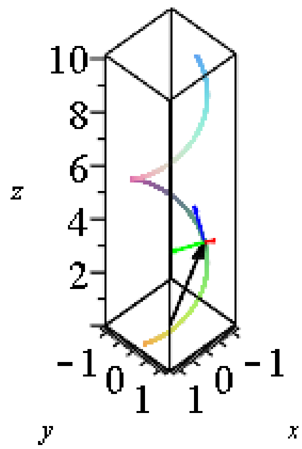


$v = 1.414213562, \quad = 0.3535533905\sqrt{2}, \quad = 0.3535533907\sqrt{2}$

Vi kan lage en animasjon når vi tegner enhetstangenten og normalvektoren langs kurven

***animate*(*BevegelseTNB*, [(*r2*, 0, 3 Pi, *n*)], *n* = 0 ..3 π, *font* = [*TIMES*, 16])**

$$n = 2.7489$$



$$v = 1.414213562, \quad = \ 0.3535533908\sqrt{2}, \quad = \ 0.3535533904\sqrt{2}$$