

# TOPOGRAPHIE

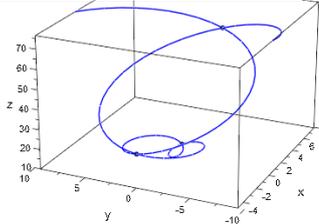
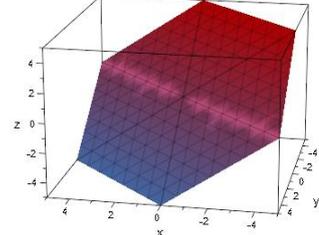
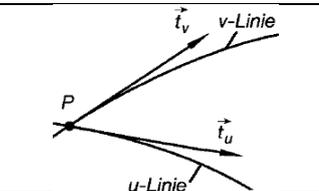
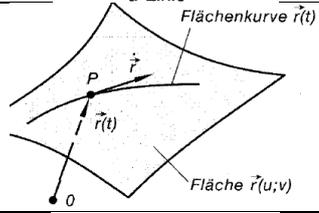
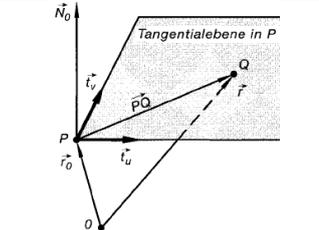
## Anwendung Topographie (MATLAB)

Import / Export			
.mar – Datei laden	load Dateiname	lädt Workspace von eine Datei	
.mat – Datei speichern	save Dateiname	speichert Workspace in eine Datei	
Plot			
Farbig 2D-Karte (Landkarte)	pcolor(A) pcolor(x,y,A)	A = 2D-Matrix x, y = Vektor mit Achsenwerten; A=2D-Matrix	
3D-Karte (Landschaft)	surf(x,y,A) surf(A) surf1(x,y,A) surf1(A)	Farbunterschiede zwischen oben/unten Farbunterschiede zwischen Sonne/Schatten-seite	
3D-Karte mit 2D-Höhenlinien am Boden	surfz(..)	Farbunterschiede zwischen oben/unten	
3D-Kontur plot	contour3(A,40)		
Attribute			
Farbskala anzeigen	colorbar('EastOutside')	'EastOutside' = Rechts aussen 'South' = Unten innen	
Farben editieren	colormap(hot)	hot, cool, jet	
Gitter ausblenden + flüssige Übergänge	shading interp		
Objekte			
Punkt zeichnen	plot(x,y,'o') plot3(x,y,z,'o')	x = Zahl mit Koordinaten	'o' = Punkt
Grösse ändern	, 'MarkerSize', 6,	im plot als hinteres Argument	
Farbe ändern	, 'MarkerFaceColor', 'y',	im plot als hinteres Argument	
Linie/Polygon in 2D-zeichnen	plot(x,y,'-')	x = Vektor mit Koordinaten	'-' = Linie
Linienstärke ändern	, 'LineWidth', 2,	im plot als hinteres Argument	
Text zeichnen	text(x,y,'Text') text(x,y,z,'Text')	x, y = Koordinaten	
Farbe ändern	, 'Color', 'r'		
Filter			
Savitzky-Golay-Filter	sgolayfilt(f,a,n)	f = Funktion	a = a'te Ableitung   n = Stufe

## MUPAD-Befehle

Funktion definieren	x:=t-->t*sin(t)	
Ableitung	diff(x(t),t)	
Vektor definieren	A := matrix([1, -1, 0]) A := matrix([1, -1, 0])	Zeilenvektor Spaltenvektor
Kreuzprodukt	linalg::crossProduct(s,v)	
Fläche darstellen	p3:=plot::Implicit3d(ax+by+cz+d=0,x=-5..5,y=-5..5,z=-5..5)	Normalendarstellung
	p4:=plot::Function3d(f(x,y),x=-5..5,y=-5..5,z=-5..5)	Funktion z=f(x,y)
	p5:=plot::Surface([x(u,v),y(u,v),z(u,v)],u=-5..5,v=-5..5)	Parameterdarstellung

**Flächen in Parameterdarstellung (MUPAD)**

<p><b>Ortsvektor einer Kurve</b></p> $\vec{r}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}, \text{ mit } t_1 \leq t \leq t_2$		<pre>x:=t--&gt;t*sin(t) y:=t--&gt;t*cos(t) z:=t--&gt;3/4*t*t+PI*PI p1:=plot::Curve3d([x(t),y(t),z(t)],t=-3*PI..3*PI) plot(p1) p2:=plot::Point3d([x,y,z]) plot(p1,p2)</pre>		
<p><b>Punkt in einer Grafik</b></p>				
<p><b>Mehrere Plots</b></p>				
<p><b>Ableitung</b></p>		<pre>x_ab:=t--&gt;diff(x(t),t) y_ab:=t--&gt;diff(y(t),t) z_ab:=t--&gt;diff(z(t),t) p1:=plot::Curve3d([x_ab(t),y_ab(t),z_ab(t)], t=-3*PI..3*PI)</pre>		
<p><b>Tangentialvektor</b></p> $\dot{\vec{r}}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix}, \text{ mit } t_1 \leq t \leq t_2$				
<p><b>Punkte</b></p> <p>A(1, -1, 0) B(1, 1, -2) C(2, -2, -1)</p>		<pre>A:=matrix([[1,-1,0]]) B:=matrix([[1,1,-2]]) C:=matrix([[2,-1,-1]])</pre>		
<p><b>Normalenvektor</b></p> $\vec{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$		<pre>n:=linalg::crossProduct(s,v) nor:=(x,y,z)--&gt;-2*x-2*y-2*z=0</pre>		
<p><b>Funktion</b></p>	$f(x,y) = x - y$	<pre>f:=(x,y)--&gt;-x-y</pre>		
<p><b>Ebene als Normalendarstellung</b></p> $ax + by + cz + d = 0$		<pre>p3:=plot::Implicit3d(nor(x,y,z),x=-5..5,___)</pre>		
<p><b>Ebene als Parameterdarstellung</b></p> $\begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \vec{OA} + u\vec{s} + v\vec{t}$		<pre>s:=B-A v:=C-A par:=A+u*s+v*t p5:=plot::Surface(par,t=-5..5,u=-5..5)</pre>		
<p><b>Ebene als Funktion</b></p>		<pre>p4:=plot::Function3d(f(x,y),x=-5..5,___)</pre>		
<p><b>Flächen im Raum</b></p>	<p>mit nichtlinearen Funktionen</p>	<pre>p6:=plot::Surface([x,y,z],t=-5..5,u=-5..5)</pre>		
<p><b>Parameterlinie einer Fläche</b></p>		<p>ein Parameter konstant setzen → v-Linie und u-Linie</p> $\vec{t}_u = \frac{d\vec{r}}{du} = \frac{dx}{du} \vec{e}_x + \frac{dy}{du} \vec{e}_y + \frac{dz}{du} \vec{e}_z$ $\vec{t}_v = \frac{d\vec{r}}{dv} = \frac{dx}{dv} \vec{e}_x + \frac{dy}{dv} \vec{e}_y + \frac{dz}{dv} \vec{e}_z$		
<p><b>Tangentenvektoren der Parameterlinie</b></p>				
<p><b>Flächenkurve</b></p> <p>Kurve auf einer Fläche</p>		$\vec{r}(t) = \vec{r}(u(t); v(t)) = \begin{pmatrix} x(u(t); v(t)) \\ y(u(t); v(t)) \\ z(u(t); v(t)) \end{pmatrix}$		
<p><b>Tangentenvektor der Flächenkurve</b></p>		<table border="1"> <tr> <td data-bbox="837 1384 1173 1485"> <p><b>direktes Ableiten</b></p> <math display="block">\dot{\vec{r}}(t) = \frac{d\vec{r}}{dt}</math> </td> <td data-bbox="1173 1384 1511 1485"> <p><b>Kettenregel</b></p> <math display="block">\dot{\vec{r}}(t) = \dot{u} \vec{t}_u + \dot{v} \vec{t}_v</math> </td> </tr> </table>	<p><b>direktes Ableiten</b></p> $\dot{\vec{r}}(t) = \frac{d\vec{r}}{dt}$	<p><b>Kettenregel</b></p> $\dot{\vec{r}}(t) = \dot{u} \vec{t}_u + \dot{v} \vec{t}_v$
<p><b>direktes Ableiten</b></p> $\dot{\vec{r}}(t) = \frac{d\vec{r}}{dt}$	<p><b>Kettenregel</b></p> $\dot{\vec{r}}(t) = \dot{u} \vec{t}_u + \dot{v} \vec{t}_v$			
<p><b>Normaleneinheitsvektor</b></p>		$N_0 = \frac{\vec{t}_u \times \vec{t}_v}{ \vec{t}_u \times \vec{t}_v }$ $\vec{N}_0 * (\vec{r} - \vec{r}_0) = 0$ $dA =  \vec{t}_u \times \vec{t}_v  du dv$		
<p><b>Tangentialebene</b></p>				
<p><b>Flächenelement dA</b></p>				

Parameterdarstellung ist nicht eindeutig!!