Computergestuetzte Mathematik zur Analysis

Lektion 10 (7. Januar)

Raumkurven (Wiederholung)

```
> restart: with(plots):
> kurve := (2 - cos(t/6))*cos(t), (2 - cos(t/6))*sin(t), t/8;
```

Achtung: kurve ist eine Folge

```
> spacecurve([ kurve, t = -6*Pi .. 6*Pi], numpoints = 300,
  thickness = 5);
> kurve := [t, k, exp(-k^2*t^2), t = -2 .. 2];
> kurvenmenge := { seq(kurve, k = 1 .. 7) };
> spacecurve(kurvenmenge, axes =  frame, thickness = 3);
> kurve := (5+cos(21*t))*cos(2*t), (5+cos(21*t))*sin(2*t), sin
  (21*t);
> spacecurve([kurve, t = 0 .. 2*Pi],numpoints=500,thickness=3);
> n := 500; j:='j'
> rgb_wert := evalf(sin(j*Pi/n)^2), 0, evalf(cos(j*Pi/n)^2);
> for j from 1 to n do;
>   p1 := subs(t = (j-1)*2*Pi/n, [kurve]);
>   p2 := subs(t = j*2*Pi/n, [kurve]);
>   pl[j] := spacecurve( [p1, p2], color = COLOR(RGB, rgb_wert),
  thickness = 3):
> od:
> display(convert(pl, set));
```

Flaechen im Raum (Wiederholung)

```
> restart:
> profil :=  cosh(t); #
> plot(cosh(t),t=-1..1);
> flaeche := [ t, cos(s)*profil, sin(s)*profil];
> plot3d(flaeche, s = 0 .. 2*Pi, t = -1 .. 1);
> plot3d(flaeche, s = 0 .. 2*Pi, t = -1 .. 1,color="DarkGreen",
  style=patchnogrid,lightmodel=light4,glossiness=0.1,viewpoint=
  "circleleft",orientation=[30,45]);
```

Partielle Ableitungen

```
> f := exp(x); with(plots):
```

```
> df := Diff(f, x);
> value(df);
> g := exp(a*x + b*y + c*z);
> dg := Diff(g, x);
> value(dg);
> d123g := Diff(g, x, y, y, z$3);
> value(d123g);
> h := (x, y, z) -> sin(a*x + b*y + c*z);
> D[2](h);
> D[1, 2, 2, 3$3](h);
> f := (x, y) -> sin(sqrt(x^2 + y^2)) * ((x-1/4)^2-(y-1/3)^2);
> p1 := plot3d(f-.05, -2 .. 2, -2 .. 2, style = surfacecontour,
  contours=30, shading = zhue):
> display(p1,orientation=[-40,50]);
> y_schnittkurve := [t, y, f(t, y), y = -2..2];
> tangente := f(1/2,-1) + D[2](f)(1/2,-1) + D[2](f)(1/2,-1)*y:
> plot([[y,f(1/2,y),y=-2..2],[y,tangente,y=-2..0]],color=[black,
  red], thickness = 3);
> y_schnitte :=spacecurve({seq(y_schnittkurve, t=-2..2,1/2)},
  color = black, thickness = 3):
> display([p1,y_schnitte],orientation=[-40,50]);
> p := <-3/2, -1, f(-3/2, -1)>;
> Dy := D[2](f)(-3/2, -1);
> y_tan := p + t.<0,1,Dy>;
> y_tan := simplify(y_tan);
> y_tan_pl := spacecurve(convert(y_tan, list), t = -1 .. 3/2,
  color = red, thickness = 5):
> display({p1,y_schnitte,y_tan_pl}, orientation=[-40,50]);
> grad := <D[1](f)(-3/2,-1),D[2](f)(-3/2,-1)>;
  ngrad := norm(grad,2):
  dgrad:= simplify(grad/ngrad):
> grad_tan := p+t.<dgrad[1],dgrad[2],ngrad>;
> grad_tan := simplify(grad_tan);
> grad_tan_pl :=spacecurve(convert(grad_tan, list), t = 0 .. 3/2,
  color = blue, thickness = 3):
> display({p1,y_schnitte,grad_tan_pl}, orientation=[90,00]);
```

Ableitungen von Vektorfunktionen

```
> restart: with(VectorCalculus):
> v := <t, t^2, t^3>;
> diff(v, t):
> with(VectorCalculus):
```

```
> diff(v, t);
> BasisFormat(false);
> dv := diff(v, t);
> with(plots):
> spacecurve(v, t = -3 .. 3,thickness=3);
```

Moebiusband

```
> restart: with(plots):
> M := <cos(t)*(1 + s*cos(t/2)),
        sin(t)*(1+s*cos(t/2)),
        s*sin(t/2)>;
  p1:= plot3d(M, t = 0 .. Pi, s=-1/2..0,color=blue):
  p2:= plot3d(M, t = 0 .. Pi, s=0..1/2,color=red):
  p3:= spacecurve(subs(s=1/2+0.02,convert(M,list)),t=0..Pi,color=
  coral,thickness=5):
  display({p1,p2,p3});
> Seele := subs(s = 0, M);
> with(VectorCalculus):
> BasisFormat(false);
> Mt := diff(Seele, t);
> Ms := diff(M, s);
> with(LinearAlgebra):
> Normale := CrossProduct(Ms, Mt);
> pl1 := plot3d(M, t = 0 .. 2*Pi, s = -1/3 .. 1/3, grid = [60,
  5], color = red):
> EinheitsNormale := simplify(Normale/Norm(Normale, 2)) assuming
  t::real:
> EinheitsNormale[1];
> flaeche := convert(Seele + s*EinheitsNormale, list);
> pl2 := plot3d(flaeche, t = 0 .. 2*Pi, s = 0 .. .4, color = s,
  numpoints = 3000, style = patchnogrid):
> with(plots):
> display({pl1, pl2}, orientation = [-78, -159]);
```