Computergestuetzte Mathematik zur Analysis

Lektion 12 (16. Januar)

```
> restart:
```

Extrema unter Nebenbedingungen

```
> with(plots):
> with(algcurves):
> with(VectorCalculus):
> BasisFormat(false):
> g := x^4 + y^4 - 1;
> NB := plot_real_curve(g, x, y);
> display(NB, scaling = constrained, thickness = 2);
> f := x + y;
> Nf := seq(solve(f = k, y),k=-1..1);
> pl_NF:= plot([Nf], x = -1 .. 1, thickness = 2): # 1. Frage
> display({NB, pl_NF}, scaling = constrained);
> grad_g := Gradient(g, [x, y]);
> grad_f := Gradient(f, [x, y]);
> GLF := grad_f - lambda*grad_g; # Gradient Lagrangefunktion
> M:={solve({GLF[1] = 0, GLF[2] = 0, g = 0}),{x,y,lambda}};
> M1 := M[1];
> M2 := M[2];
> subs(M2, x);
> allvalues(??);
> simplify(evalc([??]));
```

Alle diese Loesungen haben einen nicht-verschwindenden Imaginaerteil. Zur Kontrolle

```
> evalf(%, 2);
> M3:= M[3];
> allvalues(subs(M3,x));
> select(t -> not has(t, I), {%});
> x1 := %[1]; x2 := %%[2];
> allvalues(subs(M3, y));
> select(t -> not has(t, I), {%});
> y1 := %[1]; y2 := %%[2];
> allvalues(subs(M3, lambda));
> select(t -> not has(t, I), {%});
```

```
> lambda1 := %[1]; lambda2 := %%[2]; # 2. Frage
> it := 1:for x in [x1, x2] do;
>   for y in [y1, y2] do;
>     for lambda in [lambda1, lambda2] do;
>       print(it,x, y, lambda, GLF); it := it +1;
>     od;
>   od;
> od;
> x := 'x': y := 'y': lambda := 'lambda':
```

```
> punkt1 := {x = x1, y = y1, lambda = lambda1};
> punkt2 := {x = x2, y = y2, lambda = lambda2};
```

Test

```
> subs(punkt1, g);
> subs(punkt2, g);
> H := Hessian[VectorCalculus](f-lambda*g,[x,y]);
> subs(punkt1,H);
```

Also haben wir hier ein Minimum (Achtung: Fuer das Minimum muss nur ein Teil der Hessematrix positiv definit sein
-> Optimierung

```
> subs(punkt2,H);
```

Und hier ein Maximum

```
> wert := subs(punkt1, f);
> solve(f = wert, y);
> pl_NFs := plot(%, x = -1.2 .. .1, thickness = 2):
> display({NB, pl_NFs}, scaling = constrained);
```

Extrema unter Nebenbedingungen in 3D

```
> f :=x^2+y^2+z^2;  # 3. Frage
> g := x^2 + 2*y^2 - z^2 - 1;
> with(plots):
> gp := implicitplot3d(g, x=-2..2, y=-2..2, z=-2..2, style =
  patchcontour, shading = zhue, scaling = constrained, axes =
  boxed, numpoints = 5000):
> gp;
> Nf3o2 := implicitplot3d(f=1.5, x = -1.5 .. 1.5, y = -1.5 ..
  1.5, z= -1.5..1.5, style = patchnogrid, color = black,
  transparency = .8,numpoints=6000):
> display({Nf3o2, gp},scaling = constrained);
```

```
> grad_f := Gradient(f, [x,y,z]);
> grad_g := Gradient(g, [x, y, z]);
> GLF := grad_f - lambda*grad_g;
> Lsg := solve({GLF[1] = 0, GLF[2] = 0, GLF[3] = 0, g = 0}, {x,
  y, z, lambda});
> allvalues(Lsg[4]);
```

Loesungen sind imaginaer

```
> G1 := Lsg[1]; G2 := Lsg[2]; G3:=allvalues(Lsg[3]);
> f1 := subs(G1, f); g1:=subs(G1,g);
> f2 := subs(G2, f); g2:=subs(G2,g);
> wert12 := f1;
> ball1 := implicitplot3d(f-wert12, x = -1 .. 1, y = -1..1, z=-1.
  .1, style = patchnogrid, color = black, transparency=0.2):
> display({gp, ball1}, scaling = constrained);
> f3:=subs(G3[1],f);
> f4:=subs(G3[2],f);
> ball2 := implicitplot3d(f-1/2, x = -1 .. 1, y = -1..1, z=-1..1,
  style = patchnogrid, color = black, transparency=0.2):
> display({gp, ball2}, scaling = constrained, orientation = [-30,
  35]);
```

Zeichenkettenverarbeitung und Plotverschoenerung

```
> restart:
> for j from 1 to 3 do;
>   a[j] := "Zeile Nummer "||j;
> od:
> a[3];
> z := "Zeile Nummer ";
> z||24;
> ""||z||24;
> cat(z,24); # Verwenden sie statt || besser cat
> for j from 2 to 5 do;
>   Lsg := solve(x^j = 1, x):
>   print(Lsg[j-1]);
> od:
> for j from 2 to 5 do;
>   Lsg := solve(x^j = 1, x):
>   print(cat(j,"te Einheitswurzel:"), (Lsg[j-1]));
> od:

> f := x^2 - 1/2;
> convert(f, string);
```

```
> Beschreibung := "Graph von "||(convert(f, string));
> plot(f, x = -1 .. 1, title = Beschreibung, thickness = 2);
> Font := [TIMES, ROMAN, 24];
> plot(f, x = -1 .. 1, title = Beschreibung, font = Font,
  thickness = 2);

> Beschreibung2 := cat( Beschreibung," und ihrer Ableitung");
> Optionen := title = Beschreibung2, font = Font, titlefont =
  Font, labels = [x,y], thickness = 2;
> pl1:= plot([f, diff(f, x)], x = -1 .. 1, Optionen):
> pl1;
> with(plots):
> t1 := textplot([-.8, .2, "f"], align = {ABOVE, RIGHT}, font =
  Font):
> display({pl1, t1});
> position := -.6, subs(x = -.6, diff(f,x));
> t2 := textplot([position, "f'"], align = {BELOW, RIGHT}, font =
  Font):
> display({pl1, t1, t2}, font = Font);
> t3 := textplot3d([1, 0, 1, "Text schwebt im Raum"], font =
  Font, color = green):
> x := r*cos(theta);
> y := r*sin(theta);
> f := sqrt(1 - r^2) * (1 - sin(12*theta)/10);
> pl2 := plot3d([x, y, f], r = 0 .. 1, theta = 0 .. 2*Pi,
  lightmodel = light4, shading = zgrayscale, grid = [10, 200],
  style = patchcontour):
> display(pl2, t3);
> dateiname := "test":
> plotsetup(gif, plotoutput = dateiname);
> display(pl2, t3);
> plotsetup(window);
> display(pl2, t3);
```