# Starrkörpersimulation

## Kugel in glatter Landschaft

Alexander Artiga Gonzalez

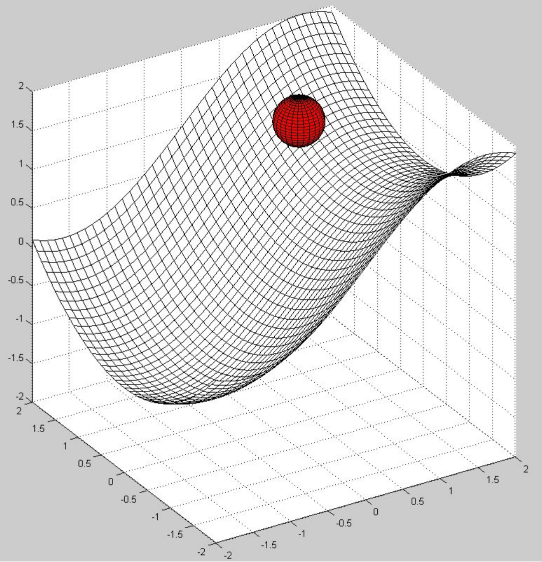
Universität Konstanz

Februar 2011

1. Problemstellung
2. Starrkörperbewegungen
3. Differentialgleichungen
4. Programmierung
5. Munte-Kaas

## Kugel in glatter Landschaft






- Glatte Landschafts-oberfläche
- Kugel kann nicht rutschen
- Keine Reibung
- Kugel ist ein Starrkörper

Abstandserhaltend:
Eine Abbildung $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ heißt abstandserhaltend, falls für alle $x, y \in \mathbb{R}^3$ gilt:
$\|L(x) - L(y)\|_2 = \|x - y\|_2$

Winkelerhaltend:
Eine Abbildung $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ heißt winkelerhaltend, falls für alle $x, y \in \mathbb{R}^3$ gilt:
$\|L(x)\|_2 \|L(y)\|_2 \langle L(x), L(y) \rangle = \|x\|_2 \|y\|_2 \langle x, y \rangle$

Orientierungserhaltend:
Eine Abbildung $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ heißt orientierungserhaltend, falls für alle $x_0, x_1, x_2, x_3 \in \mathbb{R}^3$ gilt:
$O(L(x_0), L(x_1), L(x_2), L(x_3)) = O(x_0, x_1, x_2, x_3)$ mit
$O(x_0, x_1, x_2, x_3) := sign(det(x_1 - x_0, ..., x_3 - x_0))$

Lemma 1

Sei $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ abstandserhaltend mit $L(0) = 0$. Dann ist L winkelerhaltend.

Beweis:

Für $x, y \in \mathbb{R}^3$ gilt:

$$\|L(x) - L(y)\|_2^2 = \langle L(x) - L(y), L(x) - L(y) \rangle =$$

$$= \|L(x)\|_2^2 + \|L(y)\|_2^2 - 2 \langle L(x), L(y) \rangle$$

$$\|x - y\|_2^2 = \langle x - y, x - y \rangle = \|x\|_2^2 + \|y\|_2^2 - 2 \langle x, y \rangle$$

L abstandserhaltend $\Longrightarrow \|L(x) - L(y)\|_2 = \|x - y\|_2$

Sei $y = 0$, da $L(0) = 0$ folgt: $\|L(x)\|_2 = \|x\|_2$ für alle $x \in \mathbb{R}^3$

$\Longrightarrow \langle L(x), L(y) \rangle = \langle x, y \rangle \Longrightarrow$ L ist winkelerhaltend.

Lemma 2
Sei $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ abstands- und winkelerhaltend. Dann ist L linear.

Beweis:
Sei $k \in \mathbb{N}$ beliebig. Dann gilt für beliebige $\alpha_i \in \mathbb{R}$ und $x_i \in \mathbb{R}^3$, $i = (1, .., k)$:

$$\left\langle L(\textstyle\sum_{i=1}^{k} \alpha_i x_i) - \sum_{i=1}^{k} \alpha_i L(x_i), L(u) \right\rangle =$$
$$= \left\langle L(\textstyle\sum_{i=1}^{k} \alpha_i x_i), L(u) \right\rangle - \sum_{i=1}^{k} \alpha_i \langle L(x_i), L(u) \rangle$$
$$= \left\langle \textstyle\sum_{i=1}^{k} \alpha_i x_i, u \right\rangle - \sum_{i=1}^{k} \alpha_i \langle x_i, u \rangle = 0$$

für beliebige $u \in \mathbb{R}^3 \Longrightarrow L(\sum_{i=1}^{k} \alpha_i x_i) = \sum_{i=1}^{k} \alpha_i L(x_i)$
L linear.

Lemma 3
Sei $L : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ eine abstands- und orientierungserhaltende lineare Abbildung. Dann gilt $L \in SO(3)$.

Beweis:
Es gilt:
$O(0, Le_1, Le_2, Le_3) = sign(det(L) \cdot O(0, e_1, e_2, e_3))$
Da $O(0, e_1, e_2, e_3) = \pm 1 \Rightarrow sign(detL) = 1$
$\Rightarrow L \in SO(3)$

Starrkörperbewegung in einem festen Zeitpunkt:

Sei $M : \mathbb{R}^3 \to \mathbb{R}^3$, $x \mapsto M(x)$ abstands- und orientierungserhaltend.
Sei $L(x) := M(x) - M(0) \Longrightarrow M(x) = L(x) - M(0)$.
$\Longrightarrow$ Eine Starrkörperbewegung $M(x)$ lässt sich als Rotation $(L(x))$ und Translation $(M(0))$ beschreiben.

Starrkörperbewegung:

Eine Abbildung $M : (t, x) \mapsto M(t, x) := M_t(x)$ wird als Starrkörperbewegung bezeichnet, falls sie für alle $t \in [0, T]$ abstands- und orientierungserhaltend ist. $M_t(x)$ gibt also die Position von $x$ zum Zeitpunkt $t$ an.
Zu solch einer Abbildung $M$ existieren eine Abildung $R : [0, T] \to SO(3)$ und eine Abbildung $C : [0, T] \to \mathbb{R}^3$, so dass gilt: $M_t(x) = R(t)x + c(t)$.

$$M_t(x) = R(t)x + c(t) \Longrightarrow \frac{d}{dt}M_t(x) = \dot{R}x + \dot{c}$$

$$R \in SO(3) \Longrightarrow RR^T = I$$

$$\frac{d}{dt}RR^T = 0$$

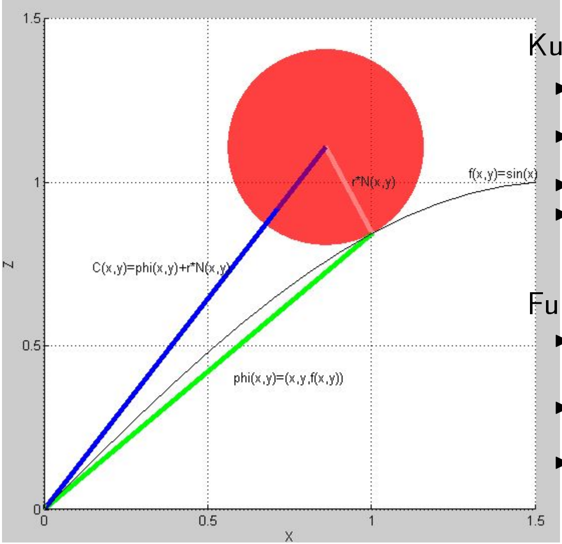
$$\Longrightarrow \dot{R}R^T + R\dot{R}^T = \dot{R}R^T + (\dot{R}R^T)^T = 0$$

$$\Longrightarrow \dot{R}R^T = -(\dot{R}R^T)^T$$

Sei $A = \dot{R}R^T$ dann folgt $A = -A^T \Longrightarrow A$ ist schiefsymmetrisch und es gilt:

$$\dot{R}(t) = A(\omega(t))R(t)$$

mit $A(\omega) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$ schiefsymmetrisch.




# Kugel:

- Referenzkugel: $B = B_r(0)$, $r > 0$
- Masse: $m(B) > 0$ (homogene Massenverteilung)
- Schwerpunkt: $\overline{x}(B) = 0$
- Trägheitstensor: $T^*_{\overline{x}}(B) = \frac{2}{5} \cdot m(B) \cdot r^2 \cdot I$

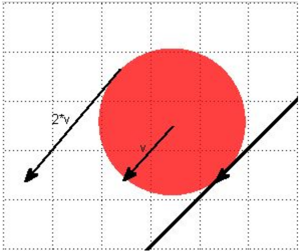
# Funktionen:

- Glatte Landschaftsoberfläche: $\varphi : \mathbb{R}^2 \to \mathbb{R}^3$ $(x, y) \mapsto (x, y, f(x, y))$
- Normalenfeld: $N : \mathbb{R}^2 \to \mathbb{R}^3$ $(x, y) \mapsto \frac{\partial_1 \varphi \times \partial_2 \varphi}{\|\partial_1 \varphi \times \partial_2 \varphi\|}$
- Mittelpunktsfunktion: $C : \mathbb{R}^2 \to \mathbb{R}^3$ $(x, y) \mapsto \varphi(x, y) + rN(x, y)$

## Differentialgleichungen



Bereits bekannt: $\dot{R}(t) = A(\omega)R(t)$
Koordinaten des Berührpunktes: $\alpha(t) \in \mathbb{R}^2$
Gesucht: $\dot{\alpha}$




Schlupffreies Rollen:

$\Longrightarrow \dot{M}_t(\alpha(t)) = \dot{R}(t)\alpha(t) + \dot{c}(t) = 0$

$\dot{c}(t) = -A(\omega(t))R(t)\alpha(t) =$
$-A(\omega(t))(b(t) - c(t)) = A(\omega(t))rN(t)$

Außerdem: $c(t) = C(\alpha(t))$
$\Longrightarrow \dot{c}(t) = C'(\alpha(t))\dot{\alpha}(t))$

Sei $E(\alpha(t)) = C'(\alpha(t))^T C'(\alpha(t))$:

$$\Longrightarrow C'(\alpha(t))^T C'(\alpha(t))\dot{\alpha}(t) = rC'(\alpha(t))^T A(\omega(t))N(\alpha(t))$$

$$\Longrightarrow \dot{\alpha}(t) = rE(\alpha(t))^{-1} C'(\alpha(t))^T A(\omega(t))N(\alpha(t))$$

Noch gesucht: $\omega$ um $\dot{R}(t) = A(\omega)R(t)$ zu lösen.

Aus Impuls-/Drehimpuls folgt:

$$(T_\alpha\omega)^{\bullet} = -m(B)A(R\dot{\alpha})A(\omega)R(\bar{x}-\alpha)$$
$$-m(B)A(R(\bar{x}-\alpha))(\dot{M}_t(\alpha))^{\bullet} + m(B)A(R(\bar{x}-\alpha))g$$

Mit $(\dot{M}_t(\alpha))^{\bullet} = 0$ (Schlupffreiheit) folgt:

$$(T_\alpha\omega)^{\bullet} = m(B)A(R(\bar{x}-\alpha))g - m(B)A(R\dot{\alpha})A(\omega)R(\bar{x}-\alpha)$$

Durch Rechnen erhält man:

$R\dot{\alpha} = rA(\omega)N(\alpha) - rN'(\alpha)$ und $R(\bar{x}-\alpha) = rN(\alpha)$

$$\Longrightarrow (T_\alpha\omega)^{\bullet} = m(B)rA(N(\alpha))g - m(B)r^2A(N'(\alpha)\dot{\alpha})A(N(\alpha))\omega$$

Mit $v(t) := T_\alpha(t)\omega(t)$ erhält man als dritte Differentialgleichung:
$\dot{v}(t) = mrA(N(\alpha(t)))g - mr^2 A(N'(\alpha(t))\dot{\alpha}(t))A(N(\alpha(t)))\omega(t)$

Differentialgleichungen

- $\dot{v}(t) = mrA(N(\alpha(t)))g - mr^2 A(N'(\alpha(t))\dot{\alpha}(t))A(N(\alpha(t)))\omega(t)$
- $\dot{\alpha}(t) = rE(\alpha(t))^{-1} C'(\alpha(t))^T A(\omega(t))N(\alpha(t))$
- $\dot{R}(t) = A(\omega)R(t)$

- GUI-Elemente in Matlab
- Lösen der Differentialgleichungen mit Runge-Kutta
- Plotten
- Speichern/Laden

Mit Hilfe von GUI-Elemente lassen sich über die Oberfläche Daten eingeben und auswerten oder einfach nur Informationen anzeigen.


```
ge.euler = uicontrol('sty','ra','ba','w','un','n','FontUnits','normalized',
'FontSize',0.5,'position',[0 .60 .08 .05],'str','Euler','ca',@eulerChecked);
```

http://www.mathworks.com/help/techdoc/ref/uicontrol.html

## Funktionen berechnen



```
function getFunctions(varargin)
   r = str2double(get(ge.radius,'string'));
   landscape = eval(['@(x,y)',get(ge.fun,'string')]);
   phi = eval(['@(x,y)','[x y landscape(x,y)]']);
   fx = diff(landscape(symx, symy), symx);
   FX = makefun(fx);
   fy = diff(landscape(symx, symy), symy);
   FY = makefun(fy);
   N = eval(['@(x,y)','[-FX(x,y) -FY(x,y) 1]./sqrt((-FX(x,y))^2+(-FY(x,y))^2+1)']);
   C = eval(['@(x,y)','phi(x,y)+r*N(x,y)']);
   dc = jacobian(C(symx, symy)+0*symx,[symx symy]);
   DC = makefun(dc);
   E = eval(['@(x,y)','DC(x,y)''*DC(x,y)']);
   dn = jacobian(N(symx, symy)+0*symx,[symx symy]);
   DN = makefun(dn);
   ref();
end

function f = makefun(s)
[m,n]=size(s);
str = '';
for i=1:m
    for j = 1:n
        str = [str,char(s(i,j)),','];
    end
    str(end)=';';
end
str = ['[',str,']'];
str = ['@(symx,symy)',str];
f = eval(str);
```

```
%Startwerte hier initialisieren
%Mögliche Tableaus:
%TAB = [1 0 ; 0 1];
TAB = [0 0 0 0 0; 0.5 0.5 0 0 0; 0.5 0 0.5 0 0; 1 0 0 1 0;0 1/6 1/3 1/3 1/6];
for i=2:n+1
    data(i,:)= data(i-1,:) + h*RKV(@f,data(i-1,:),h,TAB)';
    moveAndDisplData(data,i,t0,t);
end

function V = RKV(fhandel,yi,h,A)

%berechnet die Verfahrensfunktion V für ein RKV mit Tableau A an der Stelle
%yi von der Funktion f=fhandel mit Schrittweite h
%beachte: keine \explizites Auftreten der Zeit in f berücksichtigt!
yi = yi';
s = length(A(:,1))-1; %Stufe bestimmen

L(:,1) = fhandel(yi); %K1 initialisieren

for j = 2:s
    for i = 1:length(yi)
    z(i) = A(j,2:j)*L(i,:)';
    end
    L(:,j) = fhandel(yi+h*z'); %Iterationsschritt
end

for i = 1:length(yi)
V(i) = A(s+1,2:s+1)*L(i,:)'; %Verfahrensfunktion
end
V=V';
```

```
function [ dx ] = f(x)
%f
global A r N E m g DC DN;
R = [x(1) x(2) x(3) ; x(4) x(5) x(6) ; x(7) x(8) x(9)];
v = [x(10) x(11) x(12)]';
alpha = [x(13) x(14)];
T = R*(2/5*m*r^2*eye(3))*R'-m*r^2*A(N(alpha(1),alpha(2)))^2;
w = T\v;
dR = (A(w)*R)';
dalpha = r*(E(alpha(1),alpha(2))\(DC(alpha(1),alpha(2))'*A(w)*N(alpha(1),alpha(2))'));
dv = m*r*A(N(alpha(1),alpha(2)))*g-m*r^2*A(DN(alpha(1),alpha(2))*dalpha)*A(N(alpha(1),alpha(2)))*w;
dx = [(dR(:))' dv' dalpha']';
end
```

```
hold off;
v = str2double(get(ge.smin,'string')):0.1:str2double(get(ge.smax,'string'));
[X,Y] = meshgrid(v);
Z = landscape(X, Y);
mesh(X, Y, Z);
hold on;
NoV=26; %gerade!
[Xs Ys Zs]=sphere(NoV);
ge.ball=patch(surf2patch(Xs, Ys, Zs, Zs));
ge.refvertices=get(ge.ball,'Vertices')';
set(ge.ball,'Vertices',(r*R*ge.refvertices)'+repmat(mid,size(ge.refvertices,2),1));
```

## Speichern/Laden

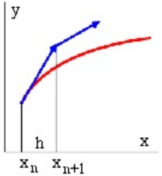


```
    %save configuration
    function saveConf(varargin)
        %write data to var
        var.r = get(ge.radius,'string');
%... fill v here
        var.max = get(ge.smax,'string');
        %save var to file
        uisave('var','examples/settings');
    end

    %load configuration
    function loadConf(varargin)
        %load var
        [FileName,PathName] = uigetfile('examples/*.mat','Choose settings');
        file=[PathName FileName];
        load(file, 'var');
        %get data from var
        set(ge.radius,'string',var.r);
%... get all values here
        set(ge.smax,'string', var.max);
        %update Functions and Plot
        opengl();zbuffer();painters();
        getFunctions();
    end
```

Rotationsmatrix ist nicht stabil. Es kann passieren, dass die Kugel sich verformt.
Grund? - Die berechnete Matrix R ist nicht mehr in $SO(3)$.

$$R(t_{n+1}) = R(t_n) + \Delta t A(\omega(t_n)) R(t_n)$$
$$= \underbrace{(I + \Delta t A(\omega(t_n)))}\, R(t)$$




Keine Drehung!

Lösungsidee: $(I + \Delta t A(\omega(t_n))) \approx \exp(\Delta t A(\omega(t_n))) \in SO(3)$
$\exp(\Delta t A(\omega(t_n))) = \sum_{k=0}^{\infty} \frac{1}{k!} (\Delta t A(\omega(t_n)))^k =$
$I + \Delta t A(\omega(t_n)) + \sum_{k=2}^{\infty} \frac{1}{k!} (\Delta t A(\omega(t_n)))^k$
$\exp(A) \exp(A)^T = \exp(A) \exp(A^T) = \exp(A) \exp(-A) = \exp(0) =$
$I \Longrightarrow \exp(A) \in SO(3)$

Berechne $R(t_{n+1})$, $w(t_{n+1})$, $\alpha(t_n+1)$ aus $R(t_n)$, $w(t_n)$, $\alpha(t_n)$
1. Schritt: Löse AWP auf $[t_n, t_{n+1}]$

- $\dot{u} = [I - \frac{1}{12}A(u)(6I + A(u))]\omega$
- $\dot{v} = mrA(N(\alpha)g - mr^2 A(N'(\alpha)\dot{\alpha})A(N(\alpha))\omega$
- $\dot{\alpha} = rE(\alpha)^{-1}C'(\alpha)^T A(\omega)N(\alpha)$

mit Anfangswerten:
$u(t_n) = 0$, $v(t_n) = T_a(t_n)\omega(t_n)$, $\alpha(t_n)$ gegeben.
$(T_a = mr^2(\frac{2}{5}I - A(N(\alpha))^2))$
2. Schritt:
$w(t_{n+1}) = T_a(t_{n+1})^{-1}v(t_{n+1})$
$R(t_{n+1}) = \exp(A(u(t_{n+1}))R(t_n)$

$\exp(A(z))$ lässt sich mit Hilfe der Rodriguez-Formel berechnen:
$\exp(A(z)) = \cos(\|z\|)I + \sin(\|z\|)A(\frac{z}{(\|z\|)}) + \frac{1-\cos(\|z\|)}{\|z\|^2} zz^T$

Beweis:
Für Potenzen von A gilt:
$A(\omega)^2 = \omega\omega^T - \|\omega\|^2 I$
$A(\omega)^{2k+1} = (-1)^k |\omega\|^2 k A(\omega), \ (k \geq 0)$
$A(\omega)^{2k} = (-1)^{k-1} |\omega\|^{2k-2} A(\omega), \ (k \geq 1)$

$\implies exp(A(z)) = \sum_{k=0}^{\infty} \frac{1}{k!} A(z)^k =$
$I + \sum_{j=1}^{\infty} \frac{1}{(2j)!} A(z)^{2j} + \sum_{j=0}^{\infty} \frac{1}{(2j+1)!} A(z)^{2j+1} =$
$1 + \sum_{j=1}^{\infty} \frac{(-1)^{j-1}}{(2j)!} \|z\|^{2j-2} A(z)^2 + \sum_{j=0}^{\infty} \frac{(-1)^j}{(2j+1)!} \|z\|^{2j} A(z) =$
$\frac{zz^T}{\|z\|^2} + \sum_{j=0}^{\infty} \frac{(-1)^{j-1}}{(2j)!} \|z\|^{2j-2} A(z)^2 + sin\|z\| A(\frac{z}{\|z\|}) =$
$\cos(\|z\|)I + \sin(\|z\|)A(\frac{z}{(\|z\|)}) + \frac{1-\cos(\|z\|)}{\|z\|^2} zz^T$