# Lorenz Wasserrad

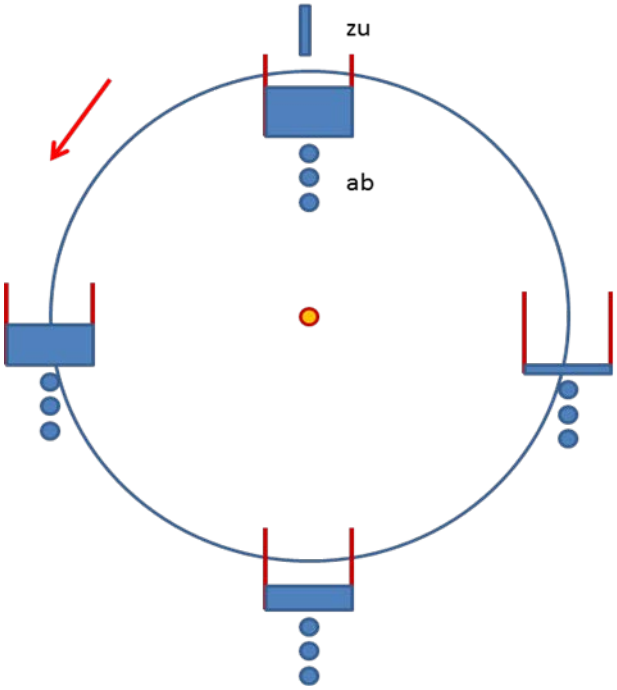
Alfred Roulier

Ein Projekt „Lorenz-Wasserrad“ bietet Gelegenheit, auf höherer Gymnasialstufe zwei auf den ersten Blick unabhängige Themen zu behandeln :

- eine Einführung in die spannende Welt des Chaos,
- eine Festigung von Elementen aus der Mechanik des starren Körpers (Winkelgeschwindigkeit, -beschleunigung, Dreh- und Trägheitsmoment).

Das Lorenz-Wasserrad ist ein mechanisches Analogon zu den Abläufen in einer Gewitterzelle. Edward N Lorenz hat 1963 diese durch drei Differentialgleichungen modelliert und bei deren Auswertung zufällig chaotisches Verhalten entdeckt. Chaos herrscht dann, einfach formuliert, wenn kleinste Abweichungen in den Ausgangsgrössen eines Systemprozesses unvorhersehbar grosse Auswirkungen auf dessen Weiterentwicklung haben. „Ein Schmetterlingsschlag auf den Azoren erzeugt einen Tornado in der Karibik“.

## Simulation




Am Umfang eines Rades sind *nk* Kessel pendelnd aufgehängt.
Oben in 12 Uhr-Stellung fliesst Wasser mit einer bestimmten Rate *zu* in die Kessel, aber alle Kessel lecken und verlieren laufend Wasser im Ausmass $zu \cdot ab$. (*zu* ist eine Rate, *ab* ein Anteil).
Die Stellung des Rades wird über den Winkel $\alpha$ zwischen einer Nullmarke und der 12 Uhr-Position beschrieben. Die Position des Kessels *j* in Funktion von $\alpha$ ist
winkel(*j*) = mod($\alpha+360\cdot(j-1)/nk,360$)

Im Dokument ***Lorenzwasserrad.tns*** simuliert das Programm ***lorenzwr()*** die Bewegung des Rades für eine Zuflussrate *zu*, einen Abflussanteil *ab* und einen Reibungskoeffizienten *reib* für eine Anzahl von *nschritt* Zeitschritten. Es gibt die Listen *plot*$\alpha$, *plot*$\omega$ und *plotzeit* aus zur Aufzeichnung in Streudiagrammen. Andererseits erzeugt das LUA Skript ***lorenz*** eine dynamische Abbildung der Radbewegung.

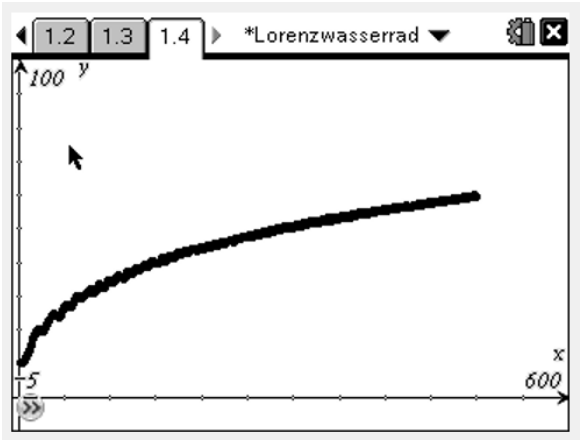
Die Programmstruktur ist wie folgt :

| |
|---|
| Definition und Initialisierung von Variablen und Listen<br>Winkelmodus „Grad“ einstellen<br>Parameter wählen, Anfangswerte setzen |
| Beginn einer Schleife über *nschritt* Zeitschritte der Länge *dt* |
| Position der nk Kessel in Funktion der Radposition α berechnen.<br>Test, ob Kessel *j* unter Zufluss steht. Wenn ja, $zu \cdot dt$ Wasser hinzufügen. Es wird darauf verzichtet, anteilig Wasser zuzuführen, wenn der Kessel nur einen Bruchteil von dt unter Zufluss steht.<br>Jedem Kessel $zu \cdot ab \cdot dt$ Wasser entziehen. |
| Dreh- , *dmom*, und Trägheitsmoment, *tmom*, des Rades bestimmen |
| Winkelbeschleunigung acc berechnen : acc = (dmom-reib$\cdot\omega$)/tmom<br>[ω=Winkelgeschwindigkeit vor Zeitschritt, reib = Reibungskoeffizient]<br>Winkelgeschwindigkeit nach Zeitschritt $\omega_{nach} = \omega_{vor} + acc \cdot dt$<br>Radposition nach Zeitschritt $\alpha_{nach} = \alpha_{vor} + \omega_{vor} \cdot dt$ |
| Ende der Schleife über die Zeit |
| Positions- und Geschwindigkeitsverlauf als Streuplot abbilden |

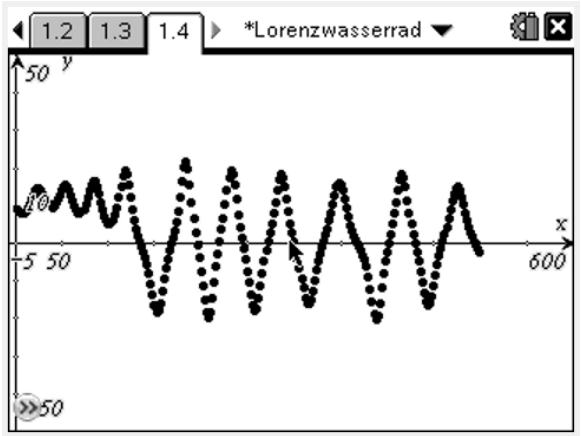
**Ergebnisse**

Wir setzen folgende feste Parameter : Anzahl Kessel nk = 8, Radius des Rades 1 m, Länge des Zeitschritts dt = 0.2 sec, Trägheitsmoment des leeren Rades mit leeren Kesseln tmom0 = 20 kg $m^2$, Seitenlänge des kubischen Kessels kdim = 0.2 m, Anfangsgeschwindigkeit des Rades $\omega_0$ = 10°/sec.

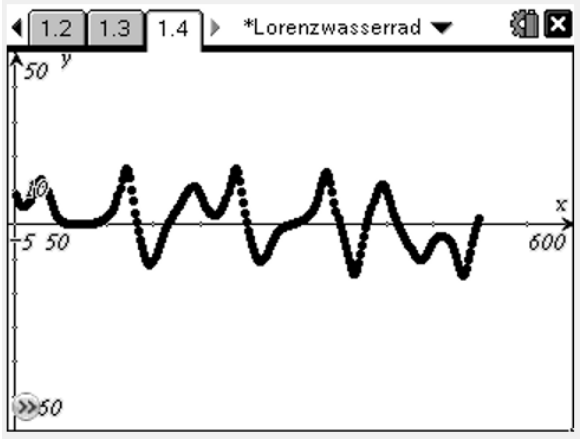
Die Zuflussrate sei 1 kg/sec, der Abflussanteil 2 %. Die Reibung steigern wir von Null aus in 3 Schritten. Es ergeben sich folgende Bilder für die Winkelgeschwindigkeit :



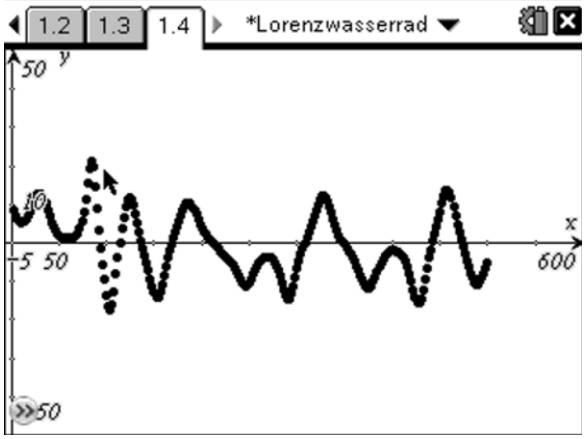

***lorenzwr****(1,.02,0,500)*
Ohne Reibung wird das Rad schubweise beschleunigt. Die Winkelgeschwindigkeit nimmt laufend zu bis ein Grenzwert erreicht ist.




***lorenzwr**(1,.02,1,500)*
Bei mässiger Reibung pendelt das Rad nach einer Einschwingphase hin und her. (Wenn der Plot durch die Nullinie geht, ändert der Drehsinn).
Der Übergang zur Oszillation ist ein Charakteristikum chaosfähiger Systeme. Man vergleiche dazu das auf der logistischen Gleichung basierende Verhulst-Diagramm.

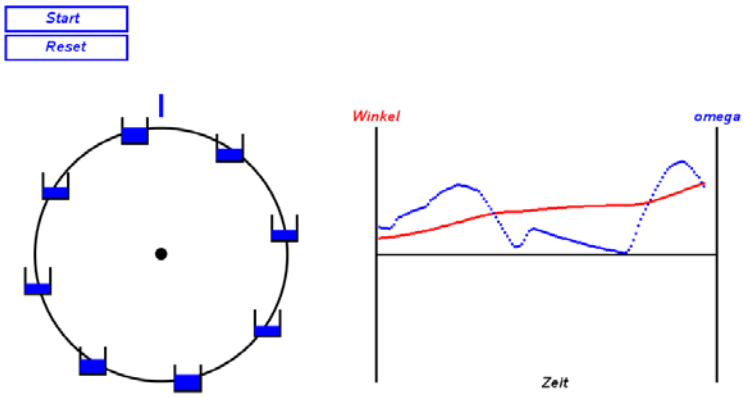



***lorenzwr**(1,.02,2,500)*
Chaos hat eingesetzt. Die Bewegung des Rades ist nun unvorhersagbar.




***lorenzwr**(1.05,.02,2,500)*
Der „Schmetterlingseffekt“ ist erkennbar : eine um 5 % höhere Zuflussrate führt zu einem gegenüber dem vorangehenden Bild deutlich anderen Verlauf.

Hier ein Standbild der mit LUA erzeugten dynamischen Visualisierung



Mehr über das Analogon unter http://www.kuen.ch/Wasserrad.php ,
mehr über E.N. Lorenz unter http://de.wikipedia.org/wiki/Edward_N._Lorenz
Bilder einer Laborausführung unter : http://www.math.cornell.edu/~lipa/mec/lesson1.html

Dr. Alfred Roulier, Neuenegg (CH)
a.roulier@bluewin.ch

## TI-nspire Programm

```
Define lorenzwr()=
Prgm
:Local tmom,dmom,inh,insum
:© Winkelmass auf Grad stellen
:setMode(2,2)
:ω:=ω0:α:=0:nk:=6:winkel:=newList(nk):inh:=((1000*kdim^(3))/(3))+newList(nk)  © Kessel sind alle zu Be-
ginn *((1)/(3)) voll
:© Simulation mit nschritt Schritten beginnen
:nschritt:=int(((simzeit)/(dt)))
:plotα:=newList(((nschritt)/(redfakt))):plotω:=newList(((nschritt)/(redfakt))):plotzeit:=newList(((nschritt)/(redfa
kt))):plotinh:=newMat(((nschritt)/(redfakt)),nk):k:=0
:© plotα, plotω, plotzeit, k : Ausgabegrössen und Index für Streuplot
:For i,1,nschritt
:© insum = Summe der Inhalte aller Kessel, dmom = Drehmoment des Rades
:   insum:=0:dmom:=0:
:   For j,1,nk
:©  Positionswinkel des Kessels j berechnen
:      winkel[j]:=mod(α+((360*(j-1))/(nk)),360)
:© Test, ob Kessel j unter Zuflussrohr; Inhalt entsprechend ändern
:      If abs(r*sin(winkel[j]))<((kdim)/(2)) and cos(winkel[j])>0 Then
:        inh[j]:=min(inh[j]+zu*(1-ab)*dt,1000*kdim^(3))
:      Else
:        inh[j]:=max(inh[j]-zu*ab*dt,0)
:      EndIf
:      insum:=insum+inh[j]
:      dmom:=dmom+9.81*inh[j]*r*sin(winkel[j])
:   EndFor
:© Dynamik nachführen
:  tmom:=tmom0+r^(2)*insum
:  acc:=((dmom-reib*ω)/(tmom))
:  ω:=ω+acc*dt
:  α:=α+ω*dt+((dt^(2))/(2))*acc
:  If mod(i,redfakt)=0 Then
:     k:=k+1
:     plotα[k]:=α:plotω[k]:=ω: plotzeit[k]:=i*dt
:     For j,1,nk
:       plotinh[k,j]:=inh[j]
:     EndFor
:  EndIf
:EndFor
:EndPrgm
```

## LUA Skript

```
-- Hier werden die Variablen in eigener function definiert. Wird in "on.construction"  und "Reset"-Prozedur
verwendet
function updateVars()
  alpha=math.pi/2+math.pi/6                -- Anfangsstellung Rad
  rr=1                                -- Radius Rad (Physik)
  kdim=.2                             -- Kesseldimension (m)
  tmom0=20                            -- Trägheitsmoment Rad + leere Kessel (kg m^2)
  nk=(var.recall("nk") or 6)          -- Anzahl Kessel
  zu=(var.recall("zu") or 1)          -- Zufluss
  ab=(var.recall("ab") or 1)          -- Abfluss
  reib=(var.recall("reib") or 1)      -- Reibung
  dt=(var.recall("dt") or 1)          -- Länge Zeitschritt
  laufzeit=0                          -- Laufzeit
  omega=math.pi/18                    -- Winkelgeschwindigkeit am Anfang
  mk={}                               -- Kesselinhalt, Kessel sind zu Beginn 1/3-voll
  for i=1,nk do
    table.insert(mk,1000*kdim^3/3)
  end
  anzp=0                              -- Anzahl Zeitschritte
  punktliste={}                       -- Liste der Ausgabegrössen in Grafik
end

-- Zur Sicherheit wird der Timer in einer function gestartet
function start()
  timer.start(dt)
end

--Konstruktion und Anwendung des Start/Stopp-Knopfs
xbutt=10 ;ybutt=10 ; lbutt=150 ; hbutt=30 ; buttN="Start"
function drawButt(gc)
  gc: drawRect(xbutt,ybutt,lbutt,hbutt)
  gc: drawString(buttN,xbutt+50,ybutt+20)
end

function inButt(X,Y)
  return (X<xbutt+lbutt)and (X>xbutt) and (Y<ybutt+hbutt)and (Y>ybutt)  -- liefert "True" wenn Bedingungen
erfüllt sind
end

function handleButt()
  if buttN=="Start" then
    start()
    buttN="Stopp"
  elseif buttN=="Stopp" then
    timer.stop() ; buttN="Start"
  end
end

--Konstruktion und Anwendung des Reset-Knopfs
xbutt1=10 ;ybutt1=45 ; lbutt=150 ; hbutt=30 ; buttN1="Reset"
function drawButt1(gc)
  gc: drawRect(xbutt1,ybutt1,lbutt,hbutt)
  gc: drawString(buttN1,xbutt1+50,ybutt1+20)
end

function inButt1(X1,Y1)
  return (X1<xbutt1+lbutt)and (X1>xbutt1) and (Y1<ybutt1+hbutt)and (Y1>ybutt1)  -- liefert "True" wenn Be-
dingungen erfüllt sind
```

```
end

function handleButt1()
   if buttN=="Start" then
    updateVars()
   elseif buttN=="Stopp" then
     timer.stop() ; buttN="Start"
     updateVars()
   end
end
function on.mouseDown(X,Y)
  if inButt(X,Y) then
     handleButt()
     platform.window:invalidate()
  elseif inButt1(X,Y) then
     handleButt1()
     platform.window:invalidate()
  end
end

function on.construction()
  hoehe=platform.window:height()
  breite=platform.window:width()
  updateVars()
end

-- falls die Bildschirmgrösse verändert wird
function on.resize(dx,dy)
   breite=dx;hoehe=dy
end

function drawAusgangslage(gc)
  r=breite/6                                    -- Radius Rad (Bild)
  gc : setPen("medium");gc: setColorRGB(0,0,0)
  gc : drawArc (0.05*breite,0.5*hoehe-r,2*r,2*r,0,360 )
  gc : fillArc (0.05*breite+0.95*r,0.5*hoehe-0.05*r,0.1*r,0.1*r,0,360)
  gc : drawLine(0.5*breite,0.5*hoehe,0.95*breite,0.5*hoehe)
  gc : drawLine(0.5*breite,0.5*hoehe-r,0.5*breite,0.5*hoehe+r)
  gc : drawLine(0.95*breite,0.5*hoehe-r,0.95*breite,0.5*hoehe+r)
  gc : setFont("sansserif","bi",12)
  gc : drawString("Zeit",0.72*breite,0.5*hoehe+1.05*r)
  gc : setColorRGB(255,0,0) ; gc : drawString ("Winkel",0.47*breite,0.5*hoehe-1.05*r)
  gc : setColorRGB(0,0,255) ; gc : drawString ("omega",0.92*breite,0.5*hoehe-1.05*r)
  gc : setPen("thick");gc:setColorRGB(0,0,255)
  gc : drawLine(0.05*breite+r,0.5*hoehe-1.25*r,0.05*breite+r,0.5*hoehe-1.1*r)
end

function drawKessel(gc)
   gc : setPen("medium")
   kessdim=r/5
   for k=1,nk do
      winkel=alpha+(k-1)*2*math.pi/nk
      xzent=0.05*breite+r+r*math.cos(winkel)
      yzent=0.5*hoehe-r*math.sin(winkel)
      gc : setColorRGB(0,0,0)
      gc : drawPolyLine({xzent-kessdim/2,yzent-kessdim/2,xzent-
kessdim/2,yzent+kessdim/2,xzent+kessdim/2,yzent+kessdim/2,xzent+kessdim/2,yzent-kessdim/2})
      fill=mk[k]/kdim^2/5
      gc : setColorRGB(0,0,255)
      gc : fillRect(xzent-kessdim/2,yzent+kessdim/2-fill,kessdim,fill)
```

```
  end
end

function neuwinkel()
  -- Drehmoment, Gesamtmasse berechnen
  insum=0 ; dmom=0                                          -- totaler Kesselinhalt ; Drehmoment
  for j=1,nk do                                          -- Zufluss/Abfluss, Trägheitsmoment, Drehmoment
berechnen
    winkel=math.fmod(alpha+(j-1)*2*math.pi/nk,2*math.pi)          -- Positionswinkel Kessel j
    if math.abs(rr*math.sin(winkel+math.pi/2))<kdim/2 and math.cos(winkel)>0 then     -- falls Kessel j oben
unter Zufuhr steht
      mk[j]=math.min(mk[j]+zu*(1-ab)*dt,1000*kdim^3)          -- Auffüllen und gleichzeitig Ver-
lust
    else
      mk[j]=math.max(mk[j]-zu*ab*dt,0)                    -- nur Verlust
    end
    insum=insum+mk[j] ; dmom=dmom+9.81*mk[j]*rr*math.sin(winkel)          -- Kesselinhalte aufad-
dieren, Drehmoment berechnen
  end
  -- Dynamik nachführen
  tmom=tmom0+rr^2*insum                                    -- neues Trägheitsmoment
  acc=(dmom-reib*omega)/tmom                               -- neue Beschleunigung
  omega=omega+acc*dt                                       -- neue Winkelgeschwindigkeit
  alpha=alpha+omega*dt+acc*dt^2/2                          -- neue Radstellung
  laufzeit=laufzeit+dt                                     -- neue Laufzeit
  anzp=anzp+1                                              -- Anzahl Zeitschritte
  table.insert(punktliste,{laufzeit,alpha,omega})          -- Liste für die Ausgabe von Teit, Win-
kel, Omega
end

function drawNeupunkte(gc)
 for i=1,anzp do
    p=punktliste[i]
    gc: setColorRGB(255,0,0)
    gc: fillRect(0.5*breite+p[1]/dt*3,0.5*hoehe-p[2]*10,3,3)
    gc: setColorRGB(0,0,255)
    gc: fillRect(0.5*breite+p[1]/dt*3,0.5*hoehe-p[3]*200,3,3)
  end
 -- Falls Grafikende erreicht, neu beginnen
  if anzp>0.45*breite/3 then
    anzp=0
    punktliste={}
    laufzeit=0
  end
end

-- wenn der der Timer getickt hat, also nach dt sec
function on.timer()
    neuwinkel()
    platform.window:invalidate()                  -- alles neu zeichnen
    start()                                  -- Timer für weitere dt sec starten
end

function on.paint(gc)
 drawAusgangslage(gc)
 drawKessel(gc)
 drawNeupunkte(gc)
 drawButt(gc)
 drawButt1(gc)
```

```
end
```