

# Einführung in die Informatik I

## Kapitel I.3: Automatisierungen von Berechnungen

Prof. Dr. Marcin Grzegorzek<sup>1</sup>

Research Group for Pattern Recognition  
[www.pr.informatik.uni-siegen.de](http://www.pr.informatik.uni-siegen.de)

Institute for Vision and Graphics  
University of Siegen, Germany



---

<sup>1</sup>Die im Rahmen dieser Lehrveranstaltung verwendeten Lernmaterialien wurden uns zum Großteil von Herrn Prof. Dr. Wolfgang Wiechert und Herrn Prof. Dr. Roland Reichardt zur Verfügung gestellt.

Ausfall der Vorlesung am 13.11

Die Vorlesung am 13.11.2013 fällt aus!

# Inhaltsverzeichnis

## **I. MATLAB-Einführung**

1. Voraussetzungen und Konventionen
2. Variablen und arithmetische Ausdrücke
- ▶ 3. Automatisierungen von Berechnungen
4. Logische Ausdrücke
5. Verzweigungen
6. Schleifen
7. Fehlersuche in Programmen
8. Funktionen
9. Arbeitsweise von Funktionen
10. Vektoren
11. Matrizen

## **II. Algorithmen**

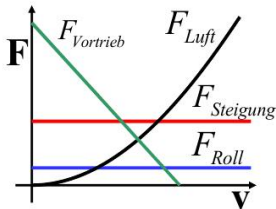
1. Suchen
2. Spezielle Suchalgorithmen
3. Sortieren
4. Rekursion und Quicksort

# Kräfte beim Fahrradfahren



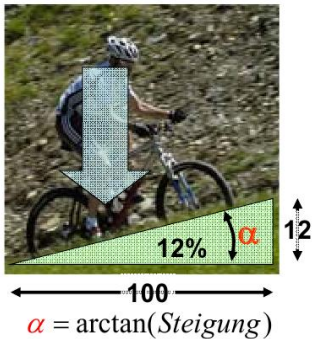
# Zielsetzung des Beispiels

- Das Beispiel zeigt eine Berechnungssequenz, wie sie im Maschinenbau häufig auftritt
- Diese Berechnung ist auf einem nicht programmierbaren Taschenrechner nur mit großem Aufwand durchführbar
- Es geht nicht um das genaue Detailverständnis aller Formeln, sondern um den grundsätzlichen Ablauf
- Ziel ist es zunächst, alle vier Kräfte als Funktion der Geschwindigkeit darzustellen
- Dazu werden physikalische Gesetzmäßigkeiten verwendet, die in den weiterführenden Vorlesungen noch detailliert behandelt werden
- Weiterhin gibt es frei wählbare Parameter (Steigung, Gewicht, Raddurchmesser, Übersetzung etc.)

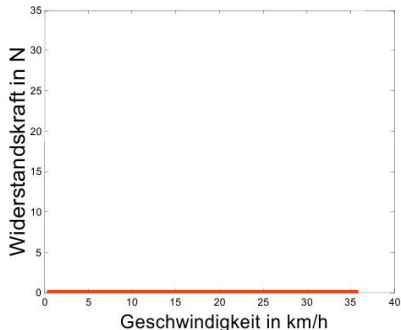


# Steigung

$$F_{\text{Steigung}} = m_{\text{ges}} g \sin(\alpha)$$

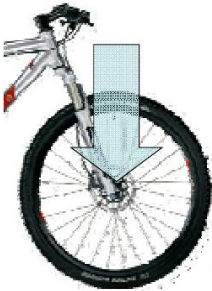


- Gesamtgewicht von Mensch und Fahrrad
- Steigung: Kraft = 0 im Flachland
- Geschwindigkeitsunabhängig

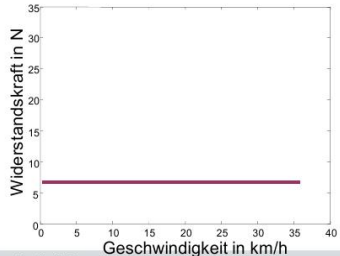


# Rollreibung

$$F_{Roll} = m_{ges} g c_r$$



- Gesamtgewicht von Mensch und Fahrrad
- Konstant:
  - Reifenprofil, Reifendruck, Untergrund, Bremse klemmt...
- Entgegengesetzt der Geschwindigkeit.



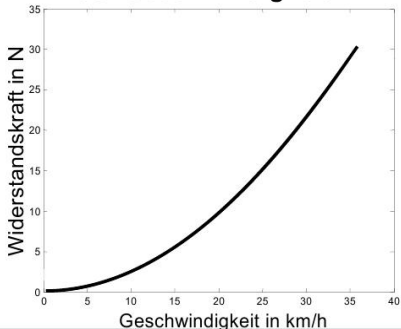


# Luftwiderstand

$$F_{Luft} = \frac{1}{2} c_w A \rho v^2$$



- $c_w$  „Windschnittigkeit“  
 $A$  Fläche, die der Wind „sieht“
- $\rho$  Dichte der Luft
- **Steigt quadratisch mit der Geschwindigkeit!**



# Wiederholen von Befehlen

- MATLAB stellt drei verschiedene Verfahren für das wiederholte Ausführen von Befehlen zur Verfügung
  - Im *Command Window* mit den Pfeiltasten die zuletzt verwendeten Befehle aufrufen.
  - Im Fenster *Command History* die zuletzt verwendeten Befehle verwenden
  - Automatische Abarbeitung mehrerer Befehle durch Skripte.

# Skripte in MATLAB (1)

- MATLAB wird primär nicht über das *Command Window* bedient, weil das auf Dauer zu umständlich ist.
- Die einzelnen Befehle können in einer Datei gespeichert und auf einmal ausgeführt werden.
- Diese Dateien müssen die Dateierweiterung „.m“ haben.
- Durch den Dateinamen wird der Befehlssatz von MATLAB erweitert.
  - Dateiname = Befehl

## Skripte in MATLAB (2)

- Neue Skripte erstellt man mit dem Menü-Befehl:
  - *File/New/M-File*
  - Innerhalb des Editors: *<Strg>-N*
- Ein neues *Editor-Fenster* wird geöffnet.
- Sinnvoller Weise speichert man das neue Skript schon vor der Fertigstellung unter dem zukünftigen Namen ab:
  - *File/Save As...*



# MATLAB: Current Directory

- In *Current Directory* Fenster werden die Dateinamen des aktuellen Verzeichnisses angezeigt.
- Dateien mit der Erweiterung „.m“ sind MATLAB Programme oder Funktionen.
- Skripte werden immer erst im aktuellen Verzeichnis gesucht



# Kommentar, Komma, Semikolon

- Kommentarzeilen beginnen mit % und werden ignoriert. Das Kommentieren hilft beim Verstehen eines Programms
  - `% Berechnung der Polynomkoeffizienten`
- Ein Semikolon nach einem Befehl unterdrückt die Ergebnisausgabe auf der Kommandozeile
  - `x=sin(2)/24; % Keine Ausgabe des Wertes`
- Ein Komma oder Zeilenwechsel nach einem Befehl führt zu einer Ergebnisausgabe auf der Kommandozeile
  - `x=5, y=6, z=7, % 5, 6, 7 werden ausgegeben`
- Kommata, Semikolons und Zeilenwechsel kann man mischen
  - `x=5, y=6; z=7 % 5, 7 werden ausgegeben`

# Matlab Script

$$F_{Roll} = m_{ges} g c_r$$

`% Rollreibung`

`cr = 0.008; % Rollreibungskoeffizient`

`g = 9.81; % Gravitationsbeschleunigung`

`m_ges = 90.2; % Gesamtmasse`

`F_Roll = m_ges * g * cr`

ohne  
Semikolon

`% Berg rauf...`

$$F_{Steigung} = m_{ges} g \sin(\alpha)$$

`Steigung = 10; % in Prozent`

`alpha = atan(Steigung / 100)`

`F_Steigung = m_ges * g * sin(alpha)`

ohne  
Semikolon

`% Konstante des Luftwiderstands ohne v2`

$$F_{Luft} = \frac{1}{2} c_w A \rho v^2$$

`v = 10.0; % Geschwindigkeit`

`cw = 0.41; % Luftwiderstandsbeiwert`

`A = 1.2463; % effektive Stirnfläche`

`rho = 1.204; % Dichte in kg/m3`

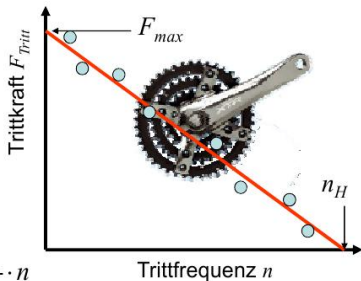
ohne  
Semikolon

`F_Luft = 1/2 * cw * A * rho * v^2`

# Trittkraft



$$F_{\text{Tritt}} = F_{\text{max}} - \frac{F_{\text{max}}}{n_H} \cdot n$$



■  $n_H$ : Höchste Trittfrequenz, Berg runter mit Rückenwind

$m_{\text{Mensch}} = 70$ ; % Gewicht des Fahrradfahrer

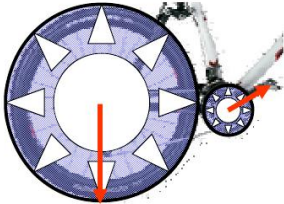
$F_{\text{max}} = m_{\text{Mensch}} \cdot g$ ; % Gewichtskraft

$n_H = 2$ ; % Höchste Trittfrequenz, aber 0 N

$F_{\text{Tritt}} = F_{\text{max}} - F_{\text{max}} / n_H \cdot n$ ;



# Vortriebskraft und Geschwindigkeit



$$\underbrace{2\pi R}_{\text{Radumfang}} \cdot i \cdot n = v$$

Radumfang

$$n = \frac{v}{2\pi R \cdot i}$$

$$F_{\text{Tritt}} = F_{\text{max}} - \frac{F_{\text{max}}}{n_H} \cdot n$$

$$F_{\text{Vortrieb}} = F_{\text{Tritt}} \frac{r}{R i}$$

- Übersetzungsverhältnis  $i$ :  
Das Hinterrad dreht sich  $i$  mal schneller als die Tretkurbel.
- Die Geschwindigkeit  $v$  errechnet sich aus dem Radumfang und der Raddrehzahl
- Vortriebskraft am Hinterrad
  - proportional zum Verhältnis der Radien (Hebelgesetz)
  - zusätzlich verringert um den Faktor  $i$
- Damit wird die Vortriebskraft zu einer Funktion der Geschwindigkeit

# Berechnung der Vortriebskraft

```
■ % Berechnung der Vortriebskraft
r      =0.15;           % Radius Tretkurbel
R      =0.35;           % Radius Rad
m_Mensch = 70;         % Gewicht des Fahrers
F_max   = m_Mensch*g;  % Gewichtskraft
nH      = 2;           % Höchste Trittfrequenz

n      = v./(2*pi*R*i);
F_Tritt = F_max - F_max / nH.*n;
F_Vortrieb = F_Tritt * r/R*i);
```

# Interaktive Eingaben

- Soll ein Skript mehrmals mit veränderlichen Parameterwerten ausgeführt werden, so kann man die Werte vom Nutzer interaktiv eingeben lassen
- Ausgabe einer Meldung im *Command Window* und warten auf eine Benutzereingabe:

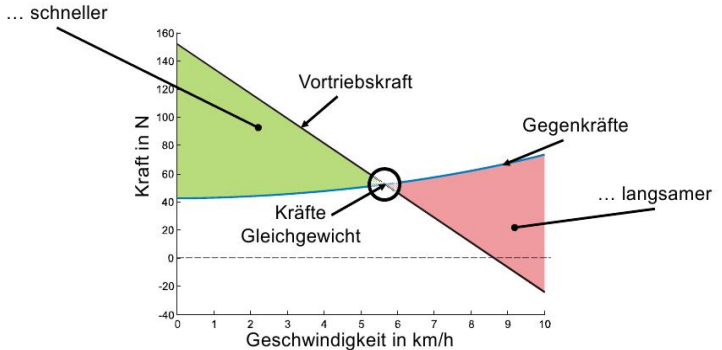
```
v=input('Geschwindigkeit : ');
```

Zwischen den Hochkommata kann ein beliebiger Text stehen.

- Ablauf der Befehlsausführung:
  - Text erscheint auf dem Bildschirm.
  - Skript wartet auf die Eingabe des Nutzers
  - Eingegebener Wert wird der Variablen  $v$  zugewiesen
- Analog erfolgt eine Ausgabe von Text und Variablenwerten im *Command Window* :

```
disp('Gesamtkraft auf das Fahrrad');  
disp(F_Ges);
```

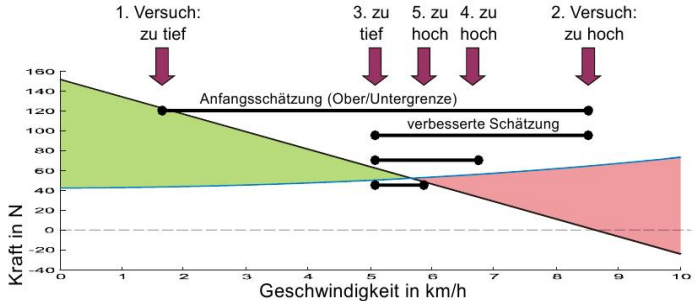
# Diagramm der Kräfte



- Am Kreuzungspunkt sind die Kräfte gleich groß:

$$F_{Vortrieb} = F_{Roll} + F_{Steigung} + F_{Luft}$$

# Einschachteln des Gleichgewichts



- Das Kräftegleichgewicht findet man interaktiv durch fortgesetzte Intervallhalbierung, beginnend mit einer Anfangsschätzung.
- Dieses Verfahren kann man mit weiteren Programmierkenntnissen vollends automatisieren.