



IWSW - Tutorium

⇒ Übung 4.2

Messdaten einlesen und verarbeiten

Dr.-Ing. Christian Vetter, Robert-Vincent Licherfeld

Institut für Prozess- und Produktionsleittechnik

Sommersemester 2017



Gliederung

Generierung der Daten

Importieren der Daten

Importieren der Daten | Alternative

Verarbeiten der Daten

Exportieren der Daten

Ergebnis in einer Funktion verwenden



Generierung der Daten

Generierung der Daten | Ultimate-Scheibe

- Standardisierte Ultimate-Scheibe
 - Masse: 0,175 [kg] | Durchmesser: 0,275 [m]

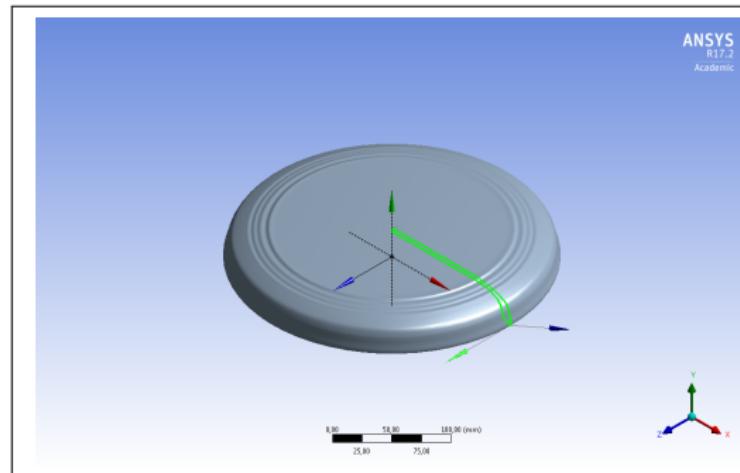


Abbildung: 3D-Modell der Ultimate-Scheibe

Generierung der Daten | Kontrollvolumen

- Kontrollvolumen um die Scheibe definieren

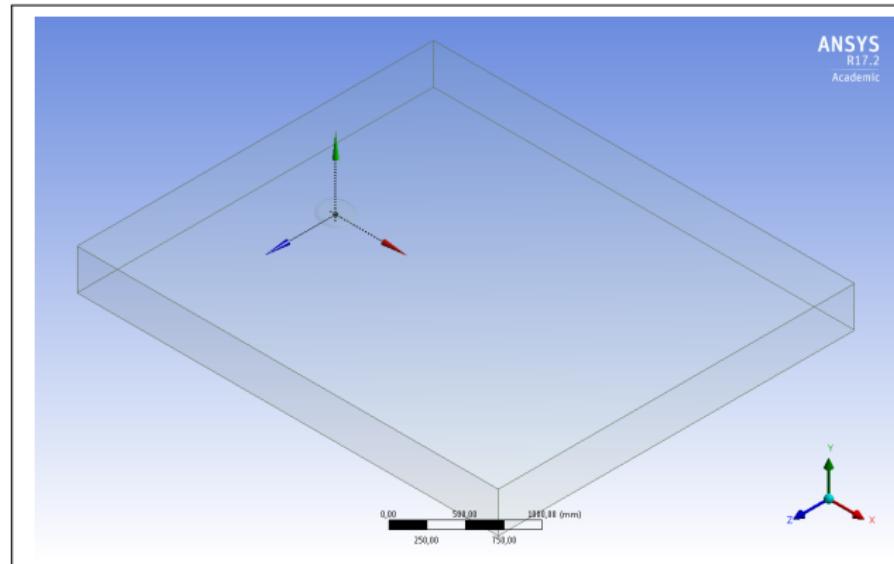


Abbildung: Kontrollvolumen zur Scheibe

Generierung der Daten | Vernetzen

- Vernetzen von Scheibe und Kontrollvolumen

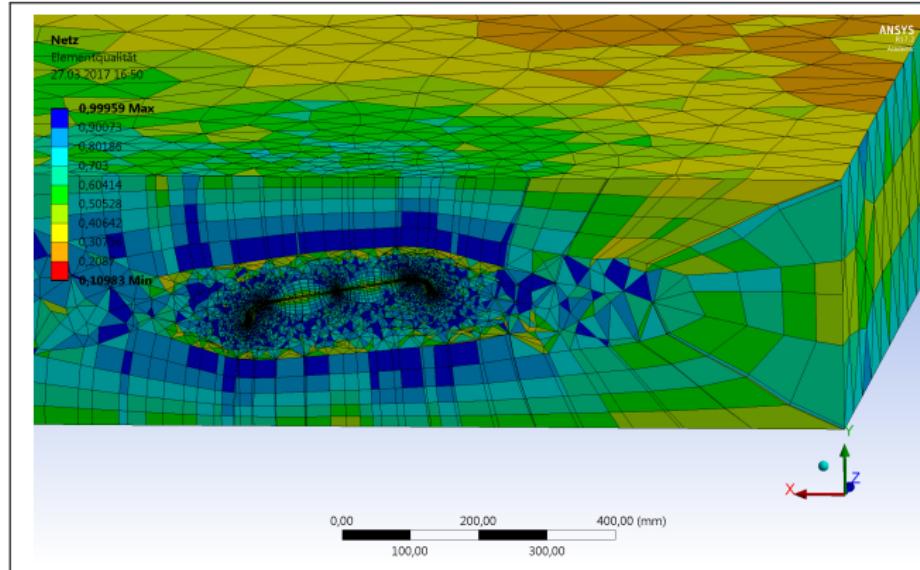


Abbildung: Vernetzung

Generierung der Daten | Setup

- Geschwindigkeit der Strömung: $v_{str} = 5 \text{ und } 7 \left[\frac{m}{s} \right]$

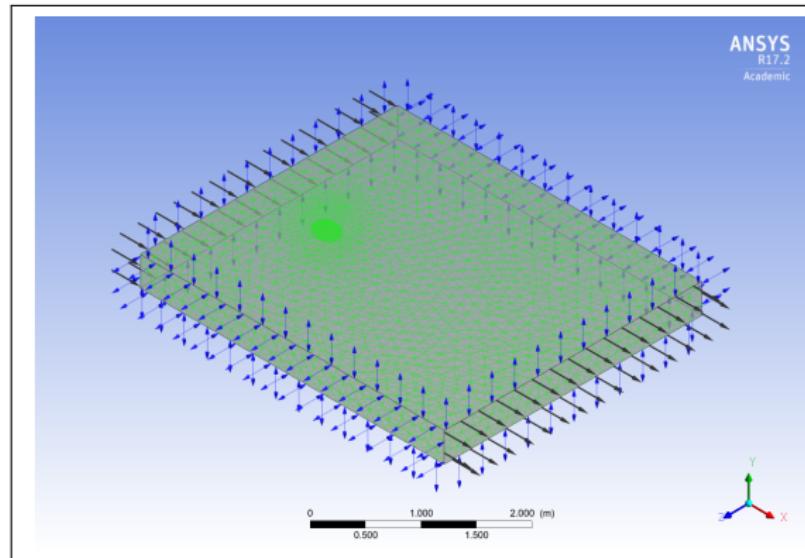


Abbildung: Setup vor Berechnung

Generierung der Daten | Strömungsberechnung

- Strömungsberechnung- und Simulation durch **ANSYS**

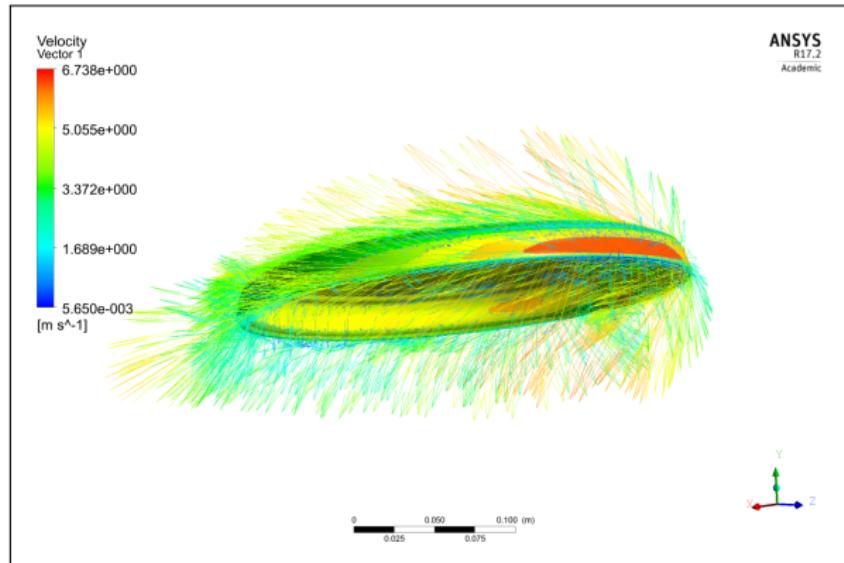


Abbildung: Vereinfachte Strömungssimulation | ANSYS 17.2



Generierung der Daten | Beiwerte

- Beiwerte und Kräfte durch ANSYS berechnet
 - Berechnung für unterschiedliche Anstellwinkel
 - Auswertung nur an diskreten Punkten

- Datei mit den Daten der Berechnung

▷ UltimateDisc.xlsx





Generierung der Daten | Beiwerte

- Beiwerte und Kräfte durch ANSYS berechnet
 - Berechnung für unterschiedliche Anstellwinkel
 - Auswertung nur an diskreten Punkten
- Datei mit den Daten der Berechnung

▷ UltimateDisc.xlsx



Generierung der Daten | Beiwerte

- Beiwerte und Kräfte durch ANSYS berechnet
 - Berechnung für unterschiedliche Anstellwinkel
 - Auswertung nur an diskreten Punkten

- Datei mit den Daten der Berechnung

▷ UltimateDisc.xlsx



Generierung der Daten | Beiwerte

- Beiwerte und Kräfte durch ANSYS berechnet
 - Berechnung für unterschiedliche Anstellwinkel
 - Auswertung nur an diskreten Punkten
- Datei mit den Daten der Berechnung

↳ UltimateDisc.xlsx





Importieren der Daten

Der try-catch-Block

- try versucht den Ausdruck auszuführen
- catch fängt eventuell auflaufende Fehler auf
 - Sammeln der Fehler auf einer Variablen (z.B. ME)
 - Anzeigen der Meldungen durch: disp(ME) ;

Listing 1: try catch Block | figure schließen

```
1 % Ausgewähltes figure schließen
2 try
3     % Versuchen figure(1) zu schließen
4     close (fg01);
5 catch ME
6     disp(ME);
7 end
```

Importieren von Daten

- Die `uiimort`-Funktion Nutzen

- Anzeigen der Verzeichnistruktur durch den '`-file`' parameter
 - » doc `uiimort` für weitere Informationen

Listing 2: Daten Importassistent

```
1 %% Übung 4.2 / Daten einlesen
2 % Dialog zum Einlesen
3 DataIN = uiimport('-file');
```

Importieren von Daten

- Die `uiimort`-Funktion Nutzen

- Anzeigen der Verzeichnistruktur durch den '`-file`' parameter
 - » doc `uiimort` für weitere Informationen

Listing 3: Daten Importassistent

```
1 %% Übung 4.2 / Daten einlesen
2 % Dialog zum Einlesen
3 DataIN = uiimport('file');
```



Importieren der Daten | Alternative

Die Funktion `uigetfile`

- `uigetfile` erstellt einen Dialog zur Dateiauswahl
- Dateiname und der absolute Pfad sind Rückgabewerte der Funktion
- » `doc uigetfile` für weitere Informationen

Listing 4: Modifiziertes Beispiel aus der Hilfe

```
1 % Dialog zur Dateiauswahl, nach dem MATLAB Hilfe-System
2 [filename, pathname] =...
3     uigetfile({'.xlsx'; '.xls'; '*.*'}, 'Datei auswählen');
4 if isequal(filename,0)
5     % Abbruch durch Benutzer anzeigen
6     disp('Dateiauswahl vom Nutzer abgebrochen!')
7 else
8     % Absoluten Pfad mit gewählter Datei anzeigen
9     disp(['Gewählte Datei: ', fullfile(pathname, filename)])
10 end
```

Die Funktion `uigetfile`

- `uigetfile` erstellt einen Dialog zur Dateiauswahl
- Dateiname und der absolute Pfad sind Rückgabewerte der Funktion
- » `doc uigetfile` für weitere Informationen

Listing 5: Modifiziertes Beispiel aus der Hilfe

```
1 % Dialog zur Dateiauswahl, nach dem MATLAB Hilfe-System
2 [filename, pathname] =...
3     uigetfile({'.xlsx'; '.xls'; '*.*'}, 'Datei auswählen');
4 if isequal(filename,0)
5     % Abbruch durch Benutzer anzeigen
6     disp('Dateiauswahl vom Nutzer abgebrochen!')
7 else
8     % Absoluten Pfad mit gewählter Datei anzeigen
9     disp(['Gewählte Datei: ', fullfile(pathname, filename)])
10 end
```

Die Funktion `uigetfile`

- `uigetfile` erstellt einen Dialog zur Dateiauswahl
- Dateiname und der absolute Pfad sind Rückgabewerte der Funktion
- » `doc uigetfile` für weitere Informationen

Listing 6: Modifiziertes Beispiel aus der Hilfe

```
1 % Dialog zur Dateiauswahl, nach dem MATLAB Hilfe-System
2 [filename, pathname] =...
3     uigetfile({'.xlsx'; '.xls'; '.*'}, 'Datei auswählen');
4 if isequal(filename,0)
5     % Abbruch durch Benutzer anzeigen
6     disp('Dateiauswahl vom Nutzer abgebrochen!')
7 else
8     % Absoluten Pfad mit gewählter Datei anzeigen
9     disp(['Gewählte Datei: ', fullfile(pathname, filename)])
10 end
```

Importieren der Daten | Alternative

- Die `xlsread` Funktion Nutzen

- Eingabeparameter: Dateiname oder Pfad mit Dateinamen
- Rückgabewerte: Datenfelder mit Zahlen, Text oder Rohdaten
- ➤ doc `xlsread` für weitere Informationen

Listing 7: Anwendung der Funktion `xlsread()`

```
1 % Excel-Datei öffnen, manuelle Auswahl der Daten durch  
2 % Parameter "-1"  
3 [num,txt,raw] = xlsread(fullfile(pathname, filename), -1);  
4 % Nur Zahlenwerte der Rückgabe nutzen  
5 DataIN = num;
```

Importieren der Daten | Alternative

- Die `xlsread` Funktion Nutzen

- Eingabeparameter: Dateiname oder Pfad mit Dateinamen
- Rückgabewerte: Datenfelder mit Zahlen, Text oder Rohdaten
- » doc `xlsread` für weitere Informationen

Listing 8: Anwendung der Funktion `xlsread()`

```
1 % Excel-Datei öffnen, manuelle Auswahl der Daten durch  
2 % Parameter "-1"  
3 [num,txt,raw] = xlsread(fullfile(pathname, filename), -1);  
4 % Nur Zahlenwerte der Rückgabe nutzen  
5 DataIN = num;
```

Importieren der Daten | Alternative

- Die `xlsread` Funktion Nutzen

- Eingabeparameter: Dateiname oder Pfad mit Dateinamen
- Rückgabewerte: Datenfelder mit Zahlen, Text oder Rohdaten
- » doc `xlsread` für weitere Informationen

Listing 9: Anwendung der Funktion `xlsread()`

```
1 % Excel-Datei öffnen, manuelle Auswahl der Daten durch  
  Parameter "-1"  
2 [num,txt,raw] = xlsread(fullfile(pathname, filename), -1);  
3 % Nur Zahlenwerte der Rückgabe nutzen  
4 DataIN = num;
```

Importdaten sortieren

- Daten werden durch `uiimport` als *struct* abgelegt
 - *struct*: Strukturiertes Datenfeld
 - Trennung der „Ebenen“ durch Punkte (vgl. Verzeichnisstruktur)
 - » doc **struct** für weitere Informationen
- Vektoren der Importdaten-Matrix auf Variablen speichern

Listing 10: Daten sortieren

```
1 % Importierte Daten sortieren
2 % Anstellwinkel aus structure array DataIN
3 aoa = DataIN.data(:, 2);
4 % Auftriebsbeiwert aus structure array DataIN
5 ca = DataIN.data(:, 5);
```

Importdaten sortieren

- Daten werden durch `uiimport` als *struct* abgelegt
 - *struct*: Strukturiertes Datenfeld
 - Trennung der „Ebenen“ durch Punkte (vgl. Verzeichnisstruktur)
 - » doc **struct** für weitere Informationen
- Vektoren der Importdaten-Matrix auf Variablen speichern

Listing 11: Daten sortieren

```
1 % Importierte Daten sortieren
2 % Anstellwinkel aus structure array DataIN
3 aoa = DataIN.data(:, 2);
4 % Auftriebsbeiwert aus structure array DataIN
5 ca = DataIN.data(:, 5);
```

Importdaten Darstellen

- Anlegen eines Fensters und Daten durch plot-Funktion darstellen
 - Titel und Achsenbeschriftung hinzufügen

Listing 12: Daten darstellen

```
1 %% Darstellung A2.4 / importierte Daten
2 % Erstellen eines figure windows und dieses auf handle referenzieren
3 fg01 = figure(1);
4 % Eigenschaften zum handle antragen, durch den Befehl set(...)
5 set(fg01,'Name','Uebung 4.2 importierte Daten','NumberTitle','on');
6 % relative Angabe [left bottom width height]
7 set(fg01,'Units','normalized','Position',[0.1 0.25 0.6 0.58]);
8
9 % Grafische Darstellung der Importierten Daten
10 plot(aoa, ca, 'o', aoa, cw, 's', 'LineWidth', 2);
```

Ergebnis | Importieren der Daten

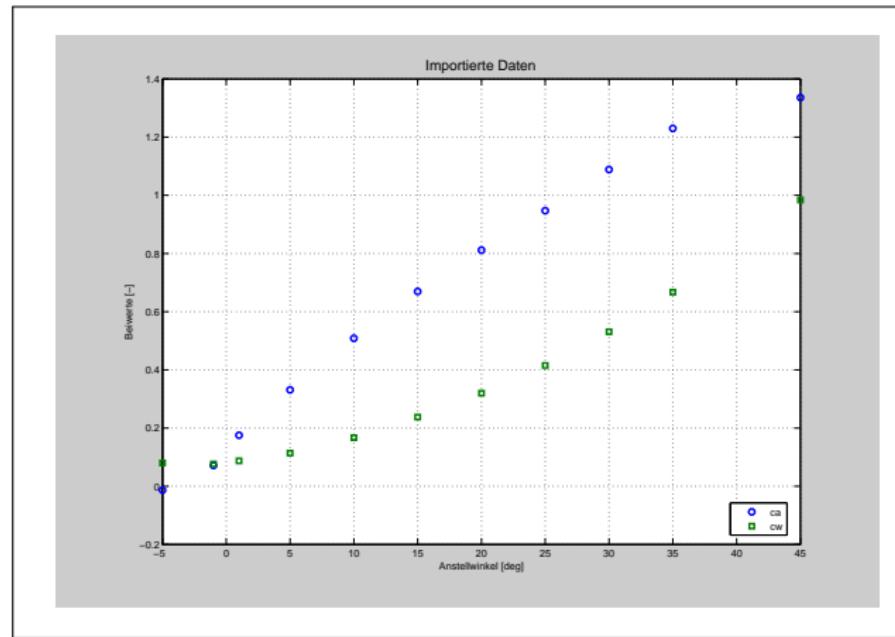


Abbildung: Darstellung der importierten Daten



Verarbeiten der Daten

Verarbeiten der Daten | Analyse | 1

- Analyse der Daten: Tools \mapsto Basic Fitting

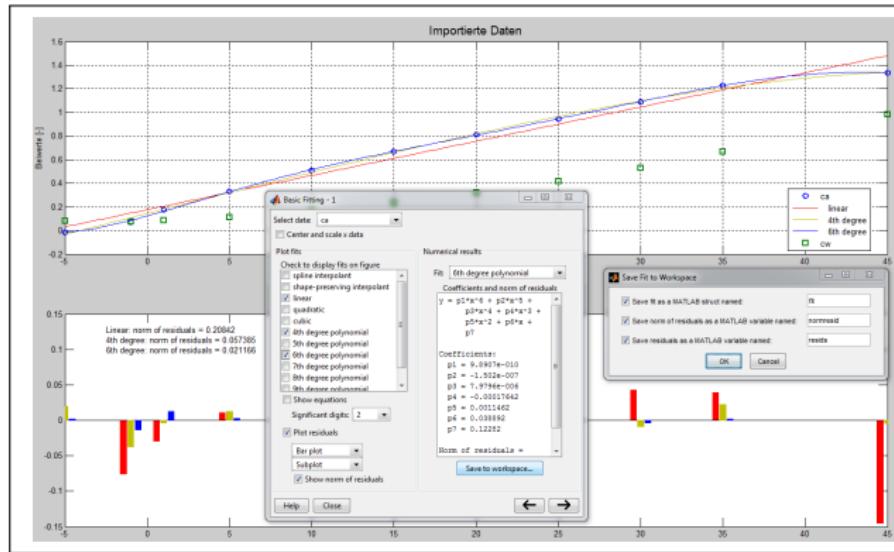


Abbildung: Analyse der Daten durch *Basic Fitting*

Verarbeiten der Daten | Analyse | 2

- Bestimmen der Koeffizienten durch die Funktion `polyfit`
 - » doc `polyfit` für weitere Informationen
- Grad des Polynoms durch die Funktion `input` erfragen

Listing 13: Koeffizienten durch `polyfit()` ermitteln

```
1 % Berechnug der Koeffizienten durch die Funktion polyfit
2 % Koeffizienten Auftriebsbeiwert
3 koeffCa = polyfit(aoa, ca, grdCa);
4
5 % Koeffizienten Widerstandsbeiwert
6 koeffCw = polyfit(aoa, cw, grdCw);
```

Verarbeiten der Daten | Analyse | 2

- Bestimmen der Koeffizienten durch die Funktion `polyfit`
 - » doc `polyfit` für weitere Informationen
- Grad des Polynoms durch die Funktion `input` erfragen

Listing 14: Koeffizienten durch `polyfit()` ermitteln

```
1 % Berechnug der Koeffizienten durch die Funktion polyfit
2 % Koeffizienten Auftriebsbeiwert
3 koeffCa = polyfit(aoa, ca, grdCa);
4
5 % Koeffizienten Widerstandsbeiwert
6 koeffCw = polyfit(aoa, cw, grdCw);
```

Verarbeiten der Daten | Analyse | 2

- Bestimmen der Koeffizienten durch die Funktion polyfit
 - » doc **polyfit** für weitere Informationen
- Grad des Polynoms durch die Funktion input erfragen

Listing 15: Koeffizienten durch polyfit() ermitteln

```
1 % Berechnug der Koeffizienten durch die Funktion polyfit
2 % Koeffizienten Auftriebsbeiwert
3 koeffCa = polyfit(aoa, ca, grdCa);
4
5 % Koeffizienten Widerstandsbeiwert
6 koeffCw = polyfit(aoa, cw, grdCw);
```



Exportieren der Daten

Exportieren der Daten | 1

- Vorbereitungen zum Export als .csv-Datei
 - Variablen benennen und Formate festlegen

Listing 16: Vorbereitungen zum Datenexport

```
1 %% Ausgabe der Koeffizienten
2 % Exportieren der Ergebnisse
3 % Name der Exportdatei im .csv-Format
4 datName = 'ExportKoeff.csv';
5 % Header für Ca schreiben
6 headrCa = sprintf('Werte-ca;');
7 % Header für Cw schreiben
8 headrCw = sprintf('Werte-cw;');
9 % Ca-Werte schreiben
10 valCa = sprintf('.12f;', koeffCa);
11 % Cw-Werte schreiben
12 valCw = sprintf('.12f;', koeffCw);
```

Exportieren der Daten | 2

- Daten in eine Datei schreiben
 - Daten zeilenweise durch Parameter '**-append**' hinzufügen

Listing 17: Daten in Datei schreiben

```
1 % In Datei schreiben
2 % Header für Ca in Datei schreiben
3 dlmwrite(datName, headrCa, 'delimiter', ',', ...
4   'newline', 'pc');
5 % Ca-Werte zur Datei anhängen
6 dlmwrite(datName, valCa, '-append', 'precision', '%.12f', ...
7   'roffset', 0, 'delimiter', ',', 'newline', 'pc');
8
9 % Header für Cw in Datei schreiben
10 dlmwrite(datName, headrCw, 'delimiter', ',', '-append', ...
11   'roffset', 2, 'newline', 'pc');
12 % Cw-Werte zur Datei anhängen
13 dlmwrite(datName, valCw, '-append', 'precision', '%.12f', ...
14   'roffset', 0, 'delimiter', ',', 'newline', 'pc');
```



Ergebnis in einer Funktion verwenden

Ergebnis in einer Funktion verwenden

- Die Funktion `flugScheibe` enthält ein vereinfachtes Flugmodell
 - Aufrufen der Funktion: `flugScheibe(caKoeff, cwKoeff);`
 - Eingangsparameter sind die Koeffizienten-Vektoren der Beiwerte
- script-Datei zur Berechnung der Flugbahn der Scheibe

▷ `flugScheibe.m`



Ergebnis in einer Funktion verwenden

- Die Funktion `flugScheibe` enthält ein vereinfachtes Flugmodell
 - Aufrufen der Funktion: `flugScheibe(caKoeff, cwKoeff);`
 - Eingangsparameter sind die Koeffizienten-Vektoren der Beiwerte
- script-Datei zur Berechnung der Flugbahn der Scheibe

▷ `flugScheibe.m`



Ergebnis | Vereinfachte Berechnung des Fluges

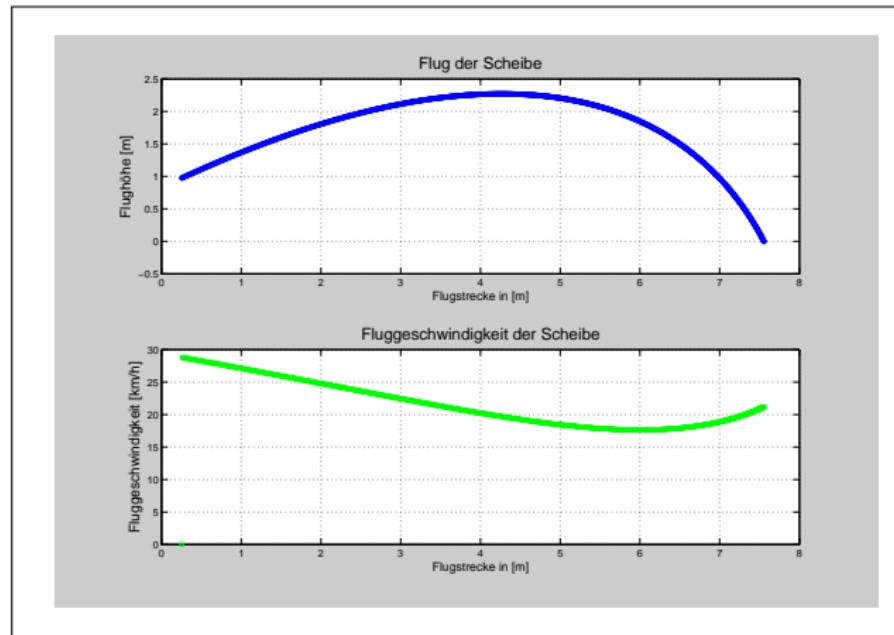


Abbildung: Flug der Scheibe

Praxistest

- Flugeigenschaften beim **Ultimate Frisbee** prüfen
- Weitere Information im **StudIP** unter „Frisbee (Ultimate)“



Abbildung: Video | Frisbee 1 | Quelle:
<http://gph.is/18mMWcl>



Abbildung: Video | Frisbee 2 | Quelle:
<http://gph.is/1jfKjeE>



Angehängte Dateien

↪ Uebung04p2.m



↪ Uebung04p2ALT.m

