

# Raumfahrttechnik

## 1. Hausaufgabe

Gruppe 14

Stefan Breit (310692)      Dimitri Sokolyuk (321116)

8. Dezember 2009

## Inhaltsverzeichnis

2.1	Aufgabe: Stufungspunkt . . . . .	3
2.2	Aufgabe: Aufstiegsbahn . . . . .	7
	<b>Listing</b>	<b>12</b>
	<b>Eidesstattliche Erklärung</b>	<b>20</b>

## Tabellenverzeichnis

1	Raketenparameter . . . . .	3
2	Konstanten . . . . .	3
3	Raketenparameter . . . . .	3
4	Massen der Komponenten . . . . .	5
5	Verbrauchte Treibstoffmassen . . . . .	6
6	Schubanstellwinkel . . . . .	8

## Abbildungsverzeichnis

1	Nutzlast über die Brenndauer der Booster . . . . .	5
2	Flughöhe über die Gesamtflugzeit . . . . .	9
3	Fluggeschwindigkeit über die Gesamtflugzeit . . . . .	10
4	Flugbahnwinkel über die Gesamtflugzeit . . . . .	11
5	Masse über die Gesamtflugzeit . . . . .	11

## 2.1 Aufgabe: Stufungspunkt

Es gilt den optimalen Stufungspunkt für eine Mission mit  $\Delta v = 12200 \text{ m/s}$  Antriebsbedarf zu ermitteln. Die Startmasse beträgt  $M_0 = 730 \text{ Mg}$ . Die Raketenparameter und Konstanten sind in den Tabellen 1 und 2 zusammengefasst.

		Feststoffbooster	Zentralstufe
Vakuumschub	$F_{vak}$	4803.3 kN	1137.9 kN
Austrittsgeschwindigkeit	$c_{e,vak}$	2663.4 m/s	4230.1 m/s
Düsenenddruck	$p_e$	$6.60 \cdot 10^5 \text{ N/m}^2$	$2.22 \cdot 10^5 \text{ N/m}^2$
Baufaktor	$\beta$	0.13	0.08
Strukturdurchmesser	$d_s$	3.1 m	5.4 m
Düsendurchmesser	$d_d$	3.0 m	2.5 m
Anzahl		2	1

Tabelle 1: Raketenparameter

Die Rakete besteht aus zwei Stufen, die parallel angeordnet sind, dabei besteht die erste Stufe aus zwei Booster und die zweite aus Zentralstufe. Der Flug lässt sich somit in zwei Flugphasen unterteilen. In der ersten Phase sind die erste und die zweite Stufe gleichzeitig aktiv. Die zweite Phase fängt an, wenn die Booster verbraucht und abgeworfen sind und dauert, bis der erforderliche Antriebsbedarf erreicht ist. Dabei ist nur die zweite Stufe aktiv.

Erdradius	$r_E = 6378.145 \text{ km}$
Erdbeschleunigung	$g_0 = 9.798 \text{ m/s}^2$
Gravitationsparameter	$\mu_E = 398600.5 \text{ km}^3/\text{s}^2$
Druck (SL)	$p_0 = 1013.25 \text{ mbar}$
Dichte (SL)	$\rho_0 = 1.214 \text{ kg/m}^3$

Tabelle 2: Konstanten

Die, für die weitere Berechnung erforderlichen Parameter sind in der Tabelle 3 zusammengefasst. Der Massenstrom der einzelnen Komponenten wird dabei aus  $F_{vak} = \dot{m} c_{e,vak}$  hergeleitet.

		Feststoffbooster	Zentralstufe
Massenstrom	$\dot{m}$	1803.4 kg/s	269.0 kg/s
Strukturquerschnitt	$A_s$	7.548 m <sup>2</sup>	22.902 m <sup>2</sup>
Düsenquerschnitt	$A_e$	7.069 m <sup>2</sup>	4.909 m <sup>2</sup>

Tabelle 3: Raketenparameter

**Optimaler Brennschluß der Booster** basiert auf folgenden Annahmen:

- Die Startmasse der 1. Phase:  $ma_{0,1} = m_0 = 730 \text{ Mg}$
- Die Brennschlußmasse der 1. Phase:  $m_{b,1} = m_{0,1} - 2m_{Fe,t,1} - m_{Fl,t,1}$ , wobei  $m_{Fe,t,1}$  und  $m_{Fl,t,1}$  entsprechen die verbrauchten Treibstoffmassen der Booster und der Zentralstufe sind.
- Die Startmasse der 2. Phase:  $m_{0,2} = m_{b,1} - 2m_{Fe,s}$ , wobei  $m_{Fe,s}$  die Strukturmasse der Booster ist.
- Die Brennschlußmasse der 2. Phase:  $m_{b,2} = m_{0,2} - m_{Fl,t,2}$ , wobei  $m_{Fl,t,2}$  die verbrauchte Treibstoffmasse der Zentralstufe ist.
- Die Nutzlast:  $m_n = m_{b,2} - m_{Fl,s}$ , wobei  $m_{Fl,s}$  die Strukturmasse der Zentralstufe ist.

Der Gesamtantriebsbedarf sätzt sich aus denen der einzelnen Phasen zusammen:

$$\Delta v = c_{e,1} \log \left( \frac{m_{0,1}}{m_{b,1}} \right) + c_{e,2} \log \left( \frac{m_{0,2}}{m_{b,2}} \right) \quad (1)$$

Der mittlere Impuls der 1. und der 2. Phase sätzt sich dabei aus den spezifischen Impulsen der ersten und der zweiten Stufen wie folgt zusammen:

$$c_{e,1} = \frac{\sum \dot{m} c_e}{\sum \dot{m}} = \frac{2 \dot{m}_{Fe,vak} c_{Fe,vak} + \dot{m}_{Fl,vak} c_{Fl,vak}}{2 \dot{m}_{Fe,vak} + \dot{m}_{Fl,vak}} = 2772.1 \text{ m/s} \quad (2)$$

$$c_{e,1} = \frac{\sum \dot{m} c_e}{\sum \dot{m}} = \frac{\dot{m}_{Fl,vak} c_{Fl,vak}}{\dot{m}_{Fl,vak}} = 4230.1 \text{ m/s} \quad (3)$$

Die verbrauchte Treibstoffmenge hängt von der Boosterbrennzeit  $t_1$  und der Gesamtbrennzeit  $t_{ges} = t_1 + t_2$  ab, wobei  $t_2$  Dauer der 2. Phase ist:

$$m_{Fe,t} = \dot{m}_{Fe} t_1 \quad (4)$$

$$m_{Fl,t} = \dot{m}_{Fl} (t_1 + t_2) \quad (5)$$

Die Bestimmung der optimalen Boosterbrennzeit erfolgt iterativ. Dabei durchläuft die die Boosterbrennzeit  $t_1$  Werte von 100 s bis 150 s. Die Wahl des Intervalls ruht auf den Erfahrungswerten der vergleichbaren Trägersystemen.

Da die Startmasse bekannt ist, lässt sich somit zu jedem Zeitpunkt des genannten Intervalls der entsprechende Antriebsvermögen und die verbrauchte Treibstoffmänge für die erste Phase bestimmen. Der Antriebsbedarf der zweiten Phase ist die Differenz aus Gesamtantriebsbedarf  $\Delta v$  und dem Antriebsvermögen der ersten Phase. Der Antriebsvermögen der zweiten Phase bestimmt seinerseits den Treibstoffverbrauch und die Brennzeit  $t_2$  der zweiten Phase und somit auch die Nutzlast.

Der Zusammenhang der Brennzeit des Boosters und der Nutzlast ist in der Abbildung 1 dargestellt. Der maximale Wert  $m_n = 5.4846$  Mg wird bei  $t_1 = 130$  s erreicht. Die Gesamtbrennzeit beträgt dabei  $t_{ges} = 670$  s.

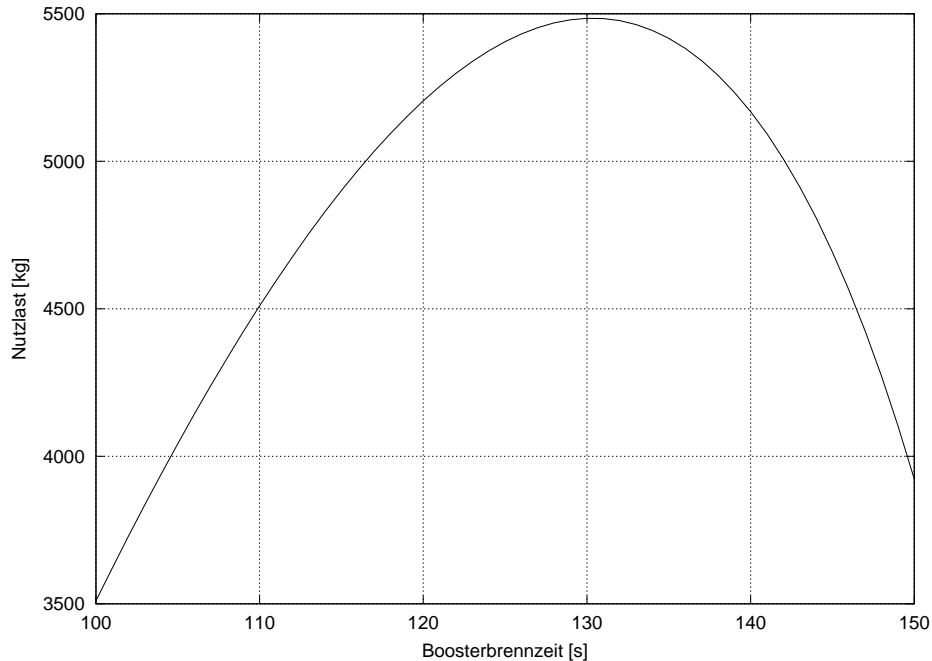


Abbildung 1: Nutzlast über die Brenndauer der Booster

**Massenverhältnisse** Die Massen der einzelnen Komponenten sind in der Tabelle 4 zusammengefasst. Somit ergeben sich folgende Massenverhältnisse:

$$\begin{aligned} \text{Strukturmassenverhältnis} \quad \sigma &= \frac{m_s}{m_0} = 0.10325 \\ \text{Treibstoffmassenverhältnis} \quad \gamma &= \frac{m_t}{m_0} = 0.88923 \\ \text{Nutzlastverhältnis} \quad \lambda &= \frac{m_n}{m_0} = 0.00751 \end{aligned}$$

Die Massengleichung der Rakete ist somit:  $\sigma + \gamma + \lambda = 1$ .

		Feststoffbooster	Zentralstufe	Gesamt
Treibstoffmasse	$m_t$	$2 \times 234.45$ Mg	180.24 Mg	649.14 Mg
Strukturmasse	$m_s$	$2 \times 30.48$ Mg	14.42 Mg	75.38 Mg
Gesamtmasse	$m_{ges}$	529.85 Mg	194.66 Mg	724.52 Mg

Tabelle 4: Massen der Komponenten

**Mittlere Impuls für die Startphase** Wie bereits in der Gleichung 3 angegeben, beträgt der mittlere Impuls für die Startphase  $I_{mittel,Start} = c_{e,1} = 2772.1 \text{ m/s}$ .

**Antriebsvermögen** Die Antriebsvermögen betragen  $\Delta v_1 = 3248.7 \text{ m/s}$  für die 1. Phase und  $\Delta v_2 = 8951.3 \text{ m/s}$  für 2. Phase entsprechend.

**Verbrauchte Treibstoffmassen** Die Übersicht der verbrauchten Treibstoffmassen ist in der Tabelle 5 dargestellt. So werden zum Brennschluß der Booster insgesamt 503.87 Mg Treibstoff verbraucht. Der Gesamtverbrauch zum absoluten Brennschluß beträgt 649.14 Mg.

	Feststoffbooster	Zentralstufe	Gesamt
Phase 1	$2 \times 234.45 \text{ Mg}$	34.97 Mg	503.87 Mg
Phase 2	—	145.27 Mg	145.27 Mg
Gesamt	468.90 Mg	180.24 Mg	649.14 Mg

Tabelle 5: Verbrauchte Treibstoffmassen

**Trägermasse zum Zeitpunkt der Stufentrennung** Die Trägermasse am Ende der ersten Phase zum Zeitpunkt der Stufentrennung setzt sich aus der Differenz der Startmasse  $m_0$  und der Masse des verbrauchten Treibstoffs und beträgt somit 261.10 Mg. Ferner wird zur Beginn der zweiten Phase die Strukturmasse der Booster abgeworfen. Die Trägermasse nach der Trennung zum Beginn der zweiten Phase beträgt somit 200.15 Mg.

**Brennzeit der Booster und der Zentralstufe** Wie bereits angegeben beträgt die nutzoptimale Brennzeit der Booster  $t_1 = 130 \text{ s}$ . Die Brennzeit der Zentralstufe entspricht der Gesamtbrennzeit und beträgt  $t_{ges} = 670 \text{ s}$ .

**Nutzlast der Trägerrakete** Wie bereits angegeben beträgt die maximale Nutzlast der Trägerrakete  $m_n = 5.4846 \text{ Mg}$ .

## 2.2 Aufgabe: Aufstiegsbahn

Es gilt die Aufstiegsbahn des Trägers in eine äquatoriale Kreisbahn von  $r_{LEO} = 500$  km Höhe zu bestimmen. Der Flug erfolgt nach einer kurzen senkrechten Aufstiegsphase mit einem konstanten Anstellwinkel des Schubes. Gesucht sind die Dauer des senkrechten Aufstieges sowie der notwendige Anstellwinkel, um den Zielorbit zu erreichen.

**Differentialgleichungssystem und Randbedingungen** Die Flugbahn des Trägers wird durch ein Differentialgleichungssystem bestimmt, welches mit Hilfe von Euler-Integrationsverfahren ausgewertet wird. Die Anfangswertbedingungen sind der Erdradius  $r_E$  als absolute Höhe des Startpunktes, Anfangsgeschwindigkeit  $v_0 = 0.005$  m/s, Flugbahnwinkel  $\gamma = 90^\circ$  und Startmasse  $m_0$ . Das Differentialgleichungssystem wird über die Gesamtflugzeit von  $t_{ges} = 670$  s ausgewertet. Für die Schrittweite  $dt$  wird der minimal geforderte Wert von 1 s gewählt.

$$\dot{r} = v \sin \gamma \quad (6)$$

$$\dot{v} = \frac{F \cos \alpha - F_w}{m} - g(r) \sin \gamma \quad (7)$$

$$\dot{\gamma} = \frac{F \sin \alpha}{mv} + \left( \frac{v}{r} - \frac{g(r)}{v} \right) \cos \gamma \quad (8)$$

$$\dot{m} = - \left( \frac{F_{Fl,vak} v_{Fl,vak}}{+} \frac{F_{Fe,vak}}{c_{Fe,vak}} \right) \quad (9)$$

Die Kreisbahngeschwindigkeit in LEO beträgt  $v_{LEO} = 7612.6$  m/s. Um den Aufstieg Treibstoffgünstiger zu gestalten, wird der Aufstieg in die Rotationsrichtung der Erde gewählt. Dabei verringert sich der notwendige Antriebsbedarf, ausgehend von dem Model der "ebenen" Erde, um die Rotationsgeschwindigkeit der Erde  $v_{rot} = 463.83$  m/s und beträgt somit  $\Delta v = v_{leo} - v_{rot} = 7148.8$  m/s.

Die Integration wird abgebrochen, wenn folgende Werte erreicht sind: die erreichte absolute Höhe  $r = r_E + r_{LEO} \pm 5$  km, die erreichte Geschwindigkeit  $v = v_{LEO} - v_{rot} \pm 5$  m/s, der erreichte Flugbahnwinkel  $\gamma = \pm 1^\circ$  und die positive Restmasse.

Das Integrationsverfahren wird ebenfalls abgebrochen, wenn die Flughöhe kleiner als Erdradius wird oder deutlich den Zielbereich übersteigt. Es wurde dafür wurde ein Grenzwert von  $r_{max} = 7 \cdot 10^3$  km der absoluten Höhe angesetzt.

Ferner wird in dem Differentialgleichungssystem folgendes berücksichtigt:

- Der Wirkungsbereich der Atmosphäre ist auf 80 km begrenzt.
- Der Boosterabwurf erfolgt zum Zeitpunkt  $t = 130$  s.
- Die Integration erfolgt in Abhängigkeit von dem Anstellwinkel des Schubes  $\alpha$  und dem Zeitpunkt der Änderung des Anstellwinkels.

**Bestimmung des Anstellwinkels** Der Anstellwinkel  $\alpha$  beträgt minimal  $-3^\circ$  und maximal  $0^\circ$ . Das negative Vorzeichen von  $\alpha$  ist durch das Differentialgleichungssystem bedingt.

Für den Zeitpunkt der Anstellwinkeländerung  $t_\alpha$  wird ein Zeitintervall, ausgehend aus den Erfahrungswerten der Vergleichbaren Trägersystemen, von 10 s bis 20 s gewählt. Dieser Bereich wird iterativ mit einer Schrittweite von 1 s durchlaufen. Der Anstellwinkel  $\alpha$  wird mit einem rekursiven Verfahren bestimmt.

Die gewählten Abbruchbedingungen des Integrationsverfahrens erlauben es die erreichbare Flughöhe auf Werte zwischen  $r_E$  und  $7 \cdot 10^3$  m einzuschranken. Die Endflughöhe zu jedem Anstellwinkel  $\alpha$  aus dem Definitionsbereich liegt somit auf jeden Fall in diesem Bereich. Diese Tatsache wird von dem rekursiven Verfahren ausgenutzt.

Die Berechnung fängt mit  $\alpha_{min} = 0^\circ$  und  $\alpha_{min} = -3^\circ$  an. Die Flugbahnhöhen werden entsprechend mit  $r_{min} = r_E$  und  $r_{max}$  angenommen. Dazu wird durch das Integrieren die Flugbahnhöhe  $r_{mid}$  zum  $\alpha_{mid} = \frac{\alpha_{min} + \alpha_{max}}{2}$  bestimmt. Das Zielorbit liegt somit entweder oberhalb oder unterhalb der errechneten Flugbahnhöhe  $r_{mid}$  und entsprechend in der oberen oder unteren Hälfte des  $\alpha$ -Intervalls. Die Berechnung wird in dem entsprechenden Bereich von  $\alpha$ -Intervall wiederholt bis der gesuchte Anstellwinkel bestimmt oder die maximale Rekursionsgrenze von 20 erreicht wird.

Die durch das Rekursivverfahren bestimmte Schubanstellwinkel zu den entsprechenden Anstellwinkeländerungszeiten sowie die dazu gehörigen Bahnparameter sind in der Tabelle 6 zusammengefasst.

$\alpha$	$t_\alpha$	$r$	$v$	$\gamma$	$m_n$	$t_{ges}$
4.2847e-02	1.0000e+01	4.9999e+02	7.1483e+03	6.9164e-01	1.5072e+04	6.2200e+02
1.2671e-01	1.4000e+01	4.9755e+02	7.1525e+03	5.2718e-01	1.5072e+04	6.2200e+02
1.5674e-01	1.5000e+01	4.9919e+02	7.1461e+03	5.7193e-01	1.5072e+04	6.2200e+02
1.9189e-01	1.6000e+01	4.9628e+02	7.1537e+03	4.2553e-01	1.5072e+04	6.2200e+02
2.2998e-01	1.7000e+01	4.9799e+02	7.1468e+03	4.6685e-01	1.5072e+04	6.2200e+02
2.7319e-01	1.8000e+01	4.9660e+02	7.1494e+03	3.7667e-01	1.5072e+04	6.2200e+02
3.2007e-01	1.9000e+01	4.9662e+02	7.1474e+03	3.4187e-01	1.5072e+04	6.2200e+02
3.7134e-01	2.0000e+01	4.9601e+02	7.1471e+03	2.7759e-01	1.5072e+04	6.2200e+02

Tabelle 6: Schubanstellwinkel

Da die gefundene Bahnparameter unterscheiden sich nur minimal von einander und



liegen innerhalb der Toleranzgrenzen liegen. Somit wird eine Bahn mit den geringsten Abweichungen von den Zielparameter als die Flugbahn ausgesucht:

Anstellwinkel	$\alpha = 0.042847^\circ$
Senkrechter Aufstieg	$t_\alpha = 10 \text{ s}$
Orbithöhe	$r = 499.99 \text{ km}$
Geschwindigkeit	$v = 7148.3 \text{ m/s}$
Flugbahnwinkel	$\gamma = 0.6916^\circ$
Nutzlast	$m_n = 15.072 \text{ Mg}$
Absoluter Brennschluß	$t_{ges} = 622 \text{ s}$

**Flughöhe** Die erreichte Flughöhe beträgt 499.99 km. Die prozentuale Abweichung von der Zielhöhe beträgt 0.002%. Der zeitlicher Verlauf der Höhenänderung über die Gesamtflugzeit ist in der Abbildung 2 dargestellt.

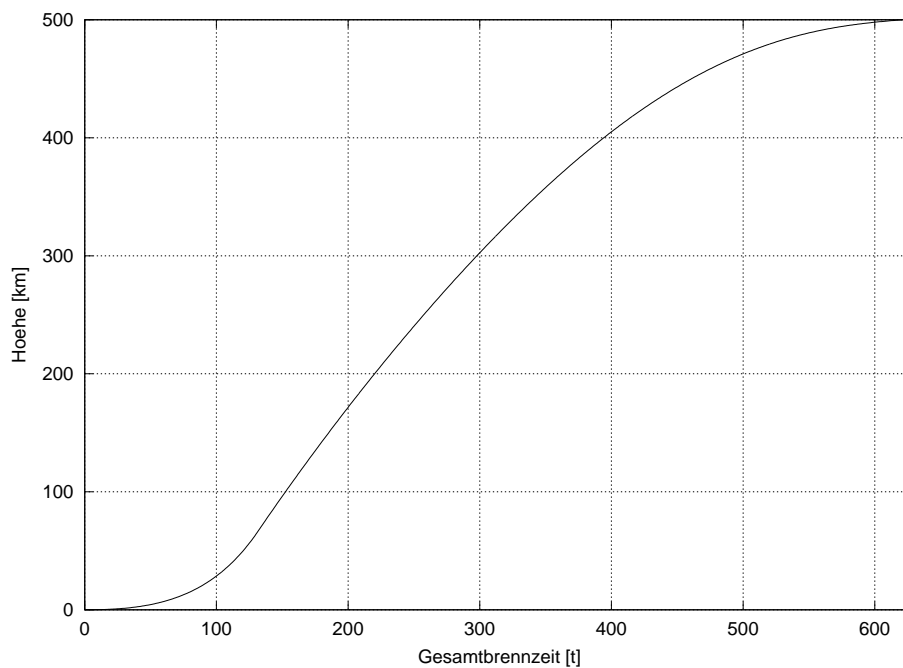


Abbildung 2: Flughöhe über die Gesamtflugzeit

**Geschwindigkeit** Die erreichte Fluggeschwindigkeit beträgt 7148.3 m/s. Die prozentuale Abweichung von der Zielgeschwindigkeit beträgt 0.006%. Der zeitlicher Verlauf der Geschwindigkeitsänderung über die Gesamtflugzeit ist in der Abbildung 3 dargestellt.

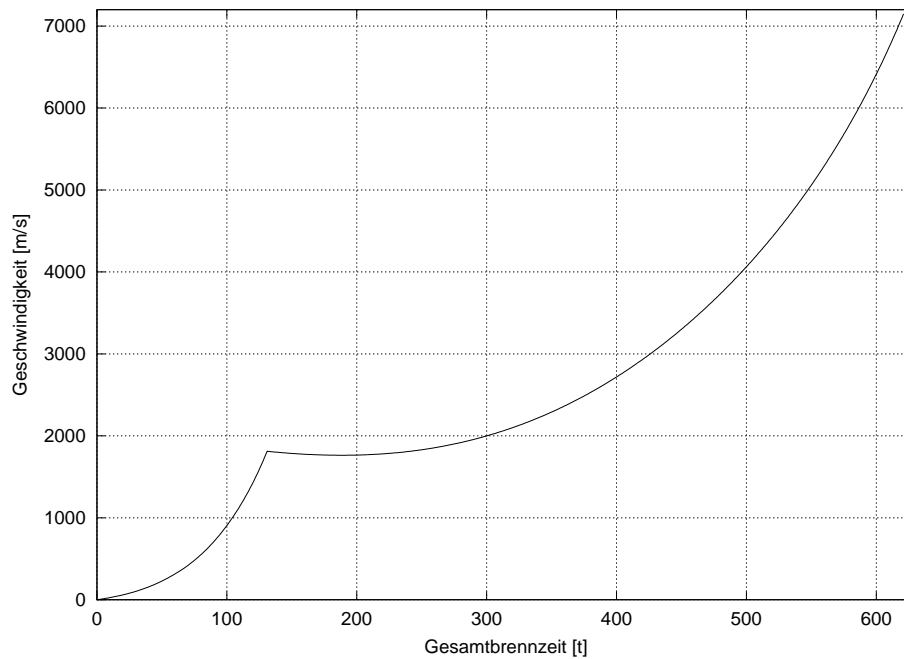


Abbildung 3: Fluggeschwindigkeit über die Gesamtflugzeit

**Flugbahnwinkel** Der erreichte Flugbahnwinkel beträgt  $0.6916^\circ$ . Der zeitlicher Verlauf der Flugbahnwinkeländerung über die Gesamtflugzeit ist in der Abbildung 4 dargestellt.

**Massen** Die Nutzlast beträgt 15.072 Mg. Der zeitlicher Verlauf der Massenänderung über die Gesamtflugzeit ist in der Abbildung 5 dargestellt.

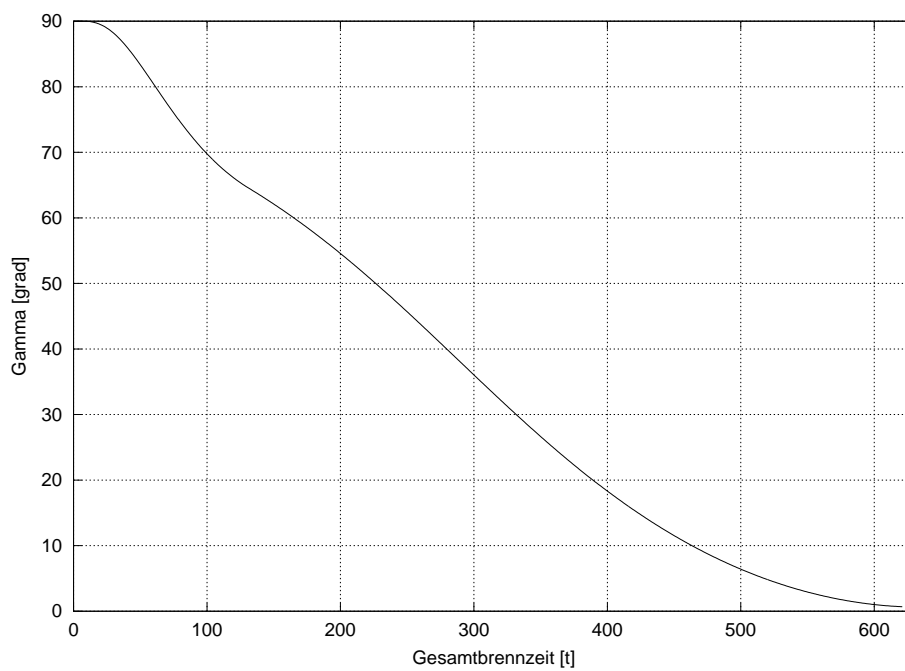


Abbildung 4: Flugbahnwinkel über die Gesamtflugzeit

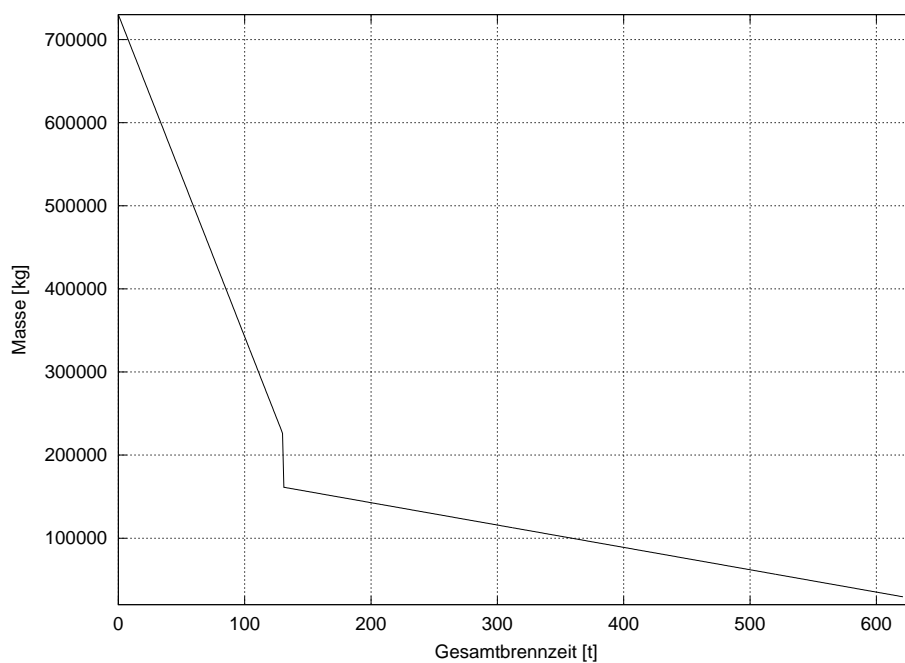


Abbildung 5: Masse über die Gesamtflugzeit

## Listing

```
1 #!/usr/bin/octave
2 #
3 # $Id: ha-02.m,v 1.7 2009/12/08 04:56:00 demon Exp demon $
4 #
5 # Grundlagen der Raumfahrttechnik WS 2009/10
6 # Hausaufgabe 2
7 #
8 # Gruppe 14
9 # Stefan Breit (310692)
10 # Dimitri Sokolyuk (321116)
11 #
12
13 clear
14 # Prevent Octave from thinking that this is a function file:
15 1;
16
17 global delta_v = 12200;      # m/s Antriebsbedarf
18 global m_0 = 730 * 10^3;    # kg Startgewicht
19 global r_leo = 500 * 10^3;  # m LEO
20 global dt = 1;             # s minimal geforderte Intergrationsschritt
21
22 # Booster
23 global booster;
24 booster.F_vak = 4803.3 * 10^3; # N Vakuumschub
25 booster.c_e_vak = 2663.4;    # m/s Austrittsgeschwindigkeit
26 booster.p_e = 6.60 * 10^5;   # N/m^2 Duesendruck
27 booster.beta = 0.13;        # 1 Baufaktor
28 booster.d = 3.1;            # m Strukturdurchmesser
29 booster.d_duse = 3.0;       # m Duesendurchmesser
30
31 # Zentralstufe
32 global main;
33 main.F_vak = 1137.9 * 10^3;  # N Vakuumschub
34 main.c_e_vak = 4230.1;     # m/s Austrittsgeschwindigkeit
35 main.p_e = 2.22 * 10^5;     # N/m^2 Duesendruck
36 main.beta = 0.08;          # 1 Baufaktor
37 main.d = 5.4;              # m Strukturdurchmesser
38 main.d_duse = 2.5;         # m Duesendurchmesser
39
40 # Konstanten
41 global r_E = 6378.145 * 10^3; # m Erdradius
42 global g_0 = 9.798;          # m/s Erdbeschleunigung @ NN
43 global mu_E = 398600.5 * 10^9; # m^3/s^2 Gravitationsparameter
44 global p_0 = 1013.25 * 10^2; # N/m^2 Luftdruck @ NN
45 global rho_0 = 1.217;       # kg/m^3 Luftdichte @ NN
46 global h_max = 80 * 10^3;   # m Atmosphergrenze
47
48 # Funktionen
49 function retval = A(d)
```

```

50 # Flaeche [m^2] aus Durchmesser [m]
51   retval = pi * d^2 / 4;
52 endfunction
53
54 function retval = m_dot(s)
55 # Massenstrom
56   retval = s.F_vak / s.c_e_vak;
57 endfunction
58
59 function retval = g(r)
60 # Erdbeschleunigung
61   global mu_E;
62   retval = mu_E ./ r.^2;
63 endfunction
64
65 function retval = c_D(Ma)
66 # Widerstandsbeiwert
67   for i = 1:1:max(size(Ma))
68     if Ma(i) <= 0.8
69       retval(i) = 0.198 + 0.0338 * Ma(i);
70     elseif Ma <= 1.0
71       retval(i) = -0.5176 + 0.9341 * Ma(i);
72     else
73       retval(i) = 0.4165 * Ma(i)^-0.603;
74     end
75   end
76 endfunction
77
78 function retval = rho(h)
79 # Dichte, input [m]
80   global rho_0 g_0 p_0 h_max;
81
82   if h < h_max
83     retval = rho_0 * exp(-rho_0 * g_0/p_0 .* h);
84   else
85     retval = 0;
86   end
87 endfunction
88
89 function retval = p(h)
90 # Druck, input [m]
91   global rho_0 g_0 p_0 h_max;
92
93   if h < h_max
94     retval = p_0 * exp(-rho_0 * g_0/p_0 .* h);
95   else
96     retval = 0;
97   end
98 endfunction
99
100 function retval = c(h)

```

```

101 # Lokale Schalgeschwindigkeit [m/s]
102 # h - Orbithoehe [m]
103 h /= 10^3; # m -> km
104 retval = 341.9862 - 5.3599 * h + 0.1419 * h^2 + 0.0012 * h^3 \
105 - 6.121 * 10^-5 * h^4 + 4.01810 * 10^-7 * h^5;
106 endfunction
107
108 #####
109 # Aufgabe 2.1
110 #####
111
112 booster.A_e = A(booster.d_duse); # m^2 Querschnittflaeche der Duese
113 booster.A_s = A(booster.d); # m^2 Querschnittflaeche der Struktur
114 booster.m_dot = m_dot(booster); # kg/s Massendurchsatz
115
116 main.A_e = A(main.d_duse); # m^2 Querschnittflaeche der Duese
117 main.A_s = A(main.d); # m^2 Querschnittflaeche der Struktur
118 main.m_dot = m_dot(main); # kg/s Massendurchsatz
119
120 # Phase 1, Gesamtaustrittsgeschwindigkeit
121 global c_e_1 = (2 * booster.m_dot * booster.c_e_vak \
122 + main.m_dot * main.c_e_vak)/(2 * booster.m_dot + main.m_dot);
123
124 # Phase 2, Gesamtaustrittsgeschwindigkeit
125 global c_e_2 = main.c_e_vak;
126
127 t_1 = [100:dt:150];
128
129 m_t_b = booster.m_dot .* t_1; # Treibstoff Booster
130 m_s_b = m_t_b * booster.beta; # Struktur Booster
131 m_ges_b = m_t_b + m_s_b; # Gesamt Booster
132
133 m_t_m = main.m_dot .* t_1; # Treibstoff Zentral Phase 1
134 m_s_m = m_t_m * main.beta; # Struktur Zentral Phase 1
135
136 m_0_1 = m_0; # Startmasse Phase 1
137 m_t_1 = 2 * m_t_b + m_t_m; # Treibstoffmasse Phase 1
138 m_b_1 = m_0_1 - m_t_1; # Brennschlussmasse Phase 1
139
140 m_0_2 = m_b_1 - 2 * m_s_b; # Startmasse Phase 2
141
142 delta_v_1 = c_e_1 * log(m_0_1 ./ m_b_1); # Antriebsbedarf Phase 1
143 delta_v_2 = delta_v - delta_v_1; # Restantriebsbedarf Phase 2
144
145 m_b_2 = m_0_2 .* exp(-delta_v_2 ./ c_e_2); # Brennschlussmasse Phase 2
146 m_t_2 = m_0_2 - m_b_2; # Treibstoffmasse Phase 2
147
148 t_2 = m_t_2 ./ main.m_dot; # Brennzeit Phase 2
149
150 m_t_m += m_t_2; # + Treibstoff Zentral Phase 2
151 m_s_m += m_t_2 * main.beta; # + Struktur Zentral Phase 2

```

```

152 m_ges_m = m_t_m + m_s_m;      # Gesamt Zentral
153
154 t_ges = t_1 + t_2;           # Gesamtbrannzeit
155
156 m_n = m_0 .- 2 * m_ges_b .- m_ges_m; # Nutzlast
157
158 [m_N_max i] = max(m_n);      # maximale Nutzlast
159 global m_N = m_N_max;
160 printf("Nutzlast: ");
161 disp(m_N);
162 global t_sep = t_1(i);      # Separationszeit
163 printf("Brennschluss Booster: ");
164 disp(t_sep);
165 global t_end = t_ges(i);    # Absoluter Brennschluss
166 printf("Absoluter Brennschluss: ");
167 disp(t_end);
168
169 booster.m_t = m_t_b(i);
170 booster.m_s = m_s_b(i);
171 booster.m_ges = m_ges_b(i);
172 main.m_t = m_t_m(i);
173 main.m_s = m_s_m(i);
174 main.m_ges = m_ges_m(i);
175
176 printf("Antriebsvermoegen 1. Phase: ");
177 disp(delta_v_1(i));
178 printf("Antriebsvermoegen 2. Phase: ");
179 disp(delta_v_2(i));
180
181 figure(1);
182 plot(t_1, m_n, "0");
183 ylabel("Nutzlast [kg]");
184 xlabel("Boosterbrennzeit [s]");
185 grid on;
186 print nutzlast.eps
187 hold off;
188
189 printf("Strukturmassenverhaeltnis: ");
190 sigma = (2 * booster.m_s + main.m_s) / m_0 # Strukturmassenverhaeltnis
191 printf("Treibstoffverhaeltnis: ");
192 gamma = (2 * booster.m_t + main.m_t) / m_0 # Treibstoffmassenverhaeltnis
193 printf("Nutzlastverhaeltnis: ");
194 lambda = m_N / m_0          # Nutzlastmassenverhaeltnis
195 printf("Massenverhaeltnis: ");
196 r = 1 / (sigma + lambda)    # Massenverhaeltnis
197 printf("sigma + gamma + lambda: ");
198 disp(sigma + gamma + lambda);
199
200 #####
201 # Aufgabe 2.2
202 #####

```

```

203
204 function rad = d2r(deg)
205 # convert degree -> radian
206   rad = deg * pi / 180;
207 endfunction
208
209 function deg = r2d(rad)
210 # convert radian -> degree
211   deg = rad * 180 / pi;
212 endfunction
213
214 # Konstanten
215 global v_rot = 2 * pi * r_E / (24 * 3600); # m/s Rotationsgeschwindigkeit am Aequator
216 global v_leo = sqrt(mu_E/(r_E + r_leo)); # m/s LEO Geschwindigkeit
217
218 # initial conditions
219 ics = [r_E 0.005 d2r(90) m_0]; # Radius, Geschwindigkeit, Winkel, Masse
220
221 function retval = abort(x)
222   global r_E r_leo v_leo v_rot;
223
224   if abs(x(1) - r_E - r_leo) < 5 * 103 \ # 5 km
225   && abs(x(2) - (v_leo - v_rot)) < 5 \ # 5 m/s
226   && abs(x(3)) < d2r(1) \ # 1 Grad
227   && x(4) >= 0
228     retval = 1;
229   else
230     retval = 0;
231   end
232
233 endfunction
234
235 function [x,t] = euler(from, step, to, startx, func, alpha, talpha)
236 # Euler
237   t = [from:step:to];
238   last = max(size(t));
239   x = zeros(max(size(t)),max(size(startx)));
240   x(1,:) = startx;
241   for i = 2:last
242     x(i,:) = x(i-1,:) + step * feval(func, x(i-1,:), t(i-1), alpha, talpha);
243     # premature abort if we already reached proper values or run out of range
244     if abort(x(i,:)) || x(i,1) < startx(1) || x(i,1) > 7 * 106
245       # strip vector length
246       x = x(1:i,:);
247       t = t(1:i);
248     return;
249   end
250 end
251 endfunction
252
253 function dx = ode(x, t, a, ta)

```



```

254 # ODE
255 global booster main r_E t_sep t_end h_max v_leo v_rot r_leo;
256
257 # rename variables for readability's sake
258 r = x(1); v = x(2); gamma = x(3); m = x(4);
259
260 alpha = 0;
261 h = r - r_E;
262 F_w = 0;
263 booster.on = 2;
264
265 # trustangle
266 if t >= ta
267     alpha = a;
268 end
269
270 # note: p(h > 80km) = 0
271 booster.F = booster.F_vak - booster.A_e * p(h);
272 main.F = main.F_vak - main.A_e * p(h);
273
274 # Brennschluss der Booster
275 if t > t_sep
276     booster.on = 0;
277 end
278
279 F = booster.on * booster.F + main.F;
280 A_s = booster.on * booster.A_s + main.A_s;
281 mach = v/c(h);
282 # note: rho(h > 80km) = 0 -> F_w = 0
283 F_w = 0.5 * c_D(mach) * rho(h) * A_s * v^2;
284
285 dr = v * sin(gamma);
286 dv = (F * cos(alpha) - F_w)/m - g(r) * sin(gamma);
287 dgamma = F * sin(alpha)/(m * v) + (v/r - g(r)/v) * cos(gamma);
288 dm = -(booster.on * booster.m_dot + main.m_dot);
289
290 # Abwurf der Booster
291 if t == t_sep
292     dm -= 2 * booster.m_s;
293 end
294
295 dx(1) = dr; dx(2) = dv; dx(3) = dgamma; dx(4) = dm;
296 endfunction
297
298 function twiddle()
299     persistent n = 0;
300     c = ["/" "-" "\" "|""];
301     printf("%s\b", c(n++ + 1));
302     n = mod(n, 4);
303 endfunction
304

```

```

305 function retval = findalpha(a, r, ics, ta)
306 # a = [0 3] alpha interval
307 # r = [10^7 0] worst case as initial condition
308 # ics
309 # ta
310
311 global dt r_E t_end r_leo;
312 persistent count = 0;
313
314 # prevent infinity loops
315 if (++count > 20)
316     retval = zeros(1,6);
317     return;
318 end
319
320 mid = (a(1) + a(2)) / 2;
321 [x,t] = euler(0, dt, t_end, ics, "ode", d2r(-mid), ta);
322 h = x(end,1);
323
324 #twiddle();
325 disp([mid ta r x(end,:)]);
326
327 if abort(x(end,:))
328     retval = [mid ta x(end,:) length(t)];
329 elseif h < ics(1) + r_leo
330     retval = findalpha([a(1) mid], [r(1) h], ics, ta);
331 else
332     retval = findalpha([mid a(2)], [h r(2)], ics, ta);
333 end
334
335 # we are done, reset counter
336 count = 0;
337 endfunction
338
339 disp("Bestimmung von alpha");
340 sol = zeros(1, 7);
341 for ta = 10:1:20
342     disp(ta);
343     z = findalpha([0 3], [10^7 0], ics, ta);
344     if z(1)
345         disp("found");
346         disp(z);
347         sol = [sol; z];
348     end
349 end
350
351 disp("Gefundene Werte");
352 sol = sol(2:end,:);
353 disp(sol);
354
355 sol(:,3) = sol(:,3) - r_E;    # Orbithoehe

```

```

356 sol(:,5) = r2d(sol(:,5)); # rad -> grad
357 sol(:,6) = sol(:,6) - main.m_s; # Nutzlast: Brennschlussmasse ohne Strukturmasse
358
359 [m_N_max i] = min(sol(:,end));
360 disp("Ausgesucher Wert");
361 disp(sol(i,:));
362
363 hold off;
364 [x,t] = euler(0, dt, t_end, ics, "ode", d2r(-sol(i,1)), sol(i,2));
365
366 printf("Gesamtflugzeit: ");
367 disp(length(t));
368
369 figure(2);
370 plot(t, (x(:,1) - r_E)/1000, "0");
371 xlabel("Gesamtbrennzeit [t]");
372 ylabel("Hoehe [km]");
373 grid on;
374 hold on;
375 print hoehe.eps;
376
377 figure(3);
378 plot(t, x(:,2), "0");
379 xlabel("Gesamtbrennzeit [t]");
380 ylabel("Geschwindigkeit [m/s]");
381 grid on;
382 hold on;
383 print geschw.eps;
384
385 figure(4);
386 plot(t, r2d(x(:,3)), "0");
387 xlabel("Gesamtbrennzeit [t]");
388 ylabel("Gamma [grad]");
389 grid on;
390 hold on;
391 print gamma.eps;
392
393 figure(5);
394 plot(t, x(:,4), "0");
395 xlabel("Gesamtbrennzeit [t]");
396 ylabel("Masse [kg]");
397 grid on;
398 hold on;
399 print masse.eps;
400
401 # EOF
402 #####

```

## Eidesstattliche Erklärung

Wir erklären hiermit an Eides Statt, dass wir die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt haben. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.