

Anhang A

Berechnungsmodell zum Fallbeispiel Rasensprenger

An warmen Sommertagen gesellt sich zur Bewässerungsfunktion des Rasensprengers noch die Nebenfunktion der Kinderbelustigung. Bei der dann angestrebten langen Nutzungsdauer gelangt zu viel Wasser auf den Rasen. Insgesamt lassen sich drei unabhängige Qualitätsmerkmale identifizieren: große Reichweite, hohe Drehzahl und geringer Wasserverbrauch. Betrachtet wird das System *Rasensprenger* ab Zuleitung hinter dem Absperrhahn.

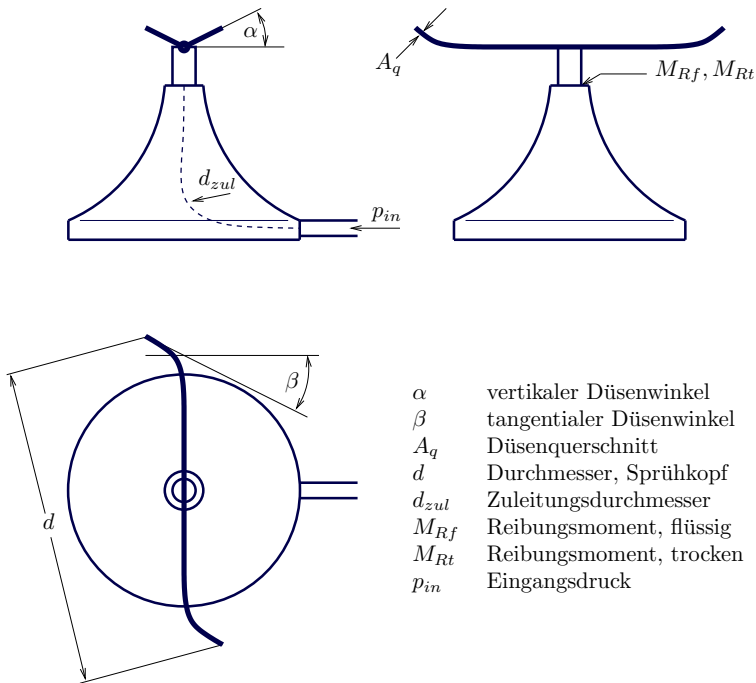


Abb. A.1 Schematische Darstellung eines Rasensprengers.

A.1 Nomenklatur

α	vertikaler Düsenwinkel
β	tangentialer Düsenwinkel
η_L	dynamische Viskosität der Luft
λ	Reibungskoeffizient
ν	kinematische Viskosität
ζ	Widerstandskoeffizient
φ	Winkel des Geschwindigkeitsvektors, bezogen auf die Horizontale
ρ, ρ_L, ρ_W	Dichte
ω	Winkelgeschwindigkeit des Rasensprengerarms
a, a_h, a_v	Beschleunigung, mit horizontaler und vertikaler Komponente
A_q	Düsenquerschnitt
A_{zul}	Querschnitt des Zulaufs
c_v	Widerstandsbeiwert, Druckverlust in der Zuleitung
d	Durchmesser des Rasensprengers
d_{Tr}	Tropfendurchmesser
d_{zul}	Zuleitungsdurchmesser, von Wasseranschluß bis Düse
F_W	Kraft auf einen Tropfen durch den Luftwiderstand
m_{Tr}	Tropfenmasse
M_A	Antriebsmoment
M_R, M_{Rf}, M_{Rt}	Reibungsmoment, mit Anteilen von flüssiger und trockener Reibung
n	Drehzahl des Rasensprengers
p_{en}	effektiv treibender Druck
p_{in}	Eingangsdruck
$\Delta p_v, \Delta p_{zul}$	Druckverluste durch Reibung
\dot{Q}	gesamter Volumenstrom
R	Radius eines Rasensprengerarms
Re	Reynoldszahl
s_h, s_v, s_{h0}, s_{v0}	hor. und vert. Komponente der Tropfenposition, mit Startwerten
v_h, v_v, v_{h0}, v_{v0}	hor. und vert. Komponente der Geschwindigkeit, mit Startwerten
$v_a, v_{at}, v_{ar}, v_{av}$	Absolutgeschwindigkeit, mit tang., radialer, vert. Komponente
$v_r, v_{at}, v_{rr}, v_{rv}$	Relativgeschwindigkeit, mit tang., radialer und vert. Komponente
v_{zul}	Strömungsgeschwindigkeit im Zulauf

A.2 Berechnung

Das bereits im Kapitel *Auswertung* verwendete Fallbeispiel wird hier näher erläutert, um dem Leser die Möglichkeit zu geben, es bei Bedarf selbst für eigene Studien zu benutzen, sozusagen als Erstanwendung. Insgesamt gibt es acht voneinander unabhängige Parameter mit dem in der Tabelle vorgeschlagenen Einstellbereich. Das

zugehörige *Octave / Matlab* Modell ist numerisch recht stabil und gestattet auch einen größeren Einstellbereich.

Parameter	Einheit	Einstellung		
		-	0	+
α	°	15	30	45
β	°	0	15	30
A_q	mm ²	2	3	4
d	mm	100	150	200
M_{Rt}	Nm	0,01	0,015	0,02
M_{Rf}	$\frac{Nm}{s}$	0,01	0,015	0,02
p_{in}	bar	1	1,5	2
d_{zul}	mm	5	7,5	10

Eine konstante Drehzahl stellt sich ein, wenn Reibungsmoment und Antriebsmoment im Gleichgewicht stehen. Das Reibungsmoment besteht aus einem konstanten Anteil, der trockenen Reibung und einem drehzahlabhängigen Anteil, der flüssigen Reibung.

$$M_R = M_{Rt} + n \cdot M_{Rf} \tag{A.1}$$

Das Antriebsmoment kommt durch den Impuls der Wassertröpfchen zustande.

$$M_A = \rho_W \dot{Q} v_{at} \cdot R \tag{A.2}$$

$$= 2\rho_W v_r A_q v_{at} \cdot R \tag{A.3}$$

Die Absolutgeschwindigkeit der Tröpfchen folgt aus dem Energiesatz,

$$v_a = \sqrt{\frac{2p_{en}}{\rho}} \tag{A.4}$$

wobei die treibende Druckdifferenz auch die Druckverluste berücksichtigt.

$$P_{en} = P_{in} - \Delta p_v - \Delta p_{zul} \tag{A.5}$$

Die Komponentenerlegung der Absolutgeschwindigkeit in tangentielle, radiale und vertikale Komponente ist nicht trivial, da die beiden Düsenwinkel die relative Ausrichtung des Wasserstrahls in Bezug zum rotierenden Rasensprenger angeben. Mit Kenntnis der Düsengeschwindigkeit läßt sich jedoch die Relativgeschwindigkeit zunächst betragsmäßig berechnen und anschließend in Komponenten zerlegen. Durch vektorielle Addition mit der Düsengeschwindigkeit folgt daraus dann die gesuchte Absolutgeschwindigkeit als vollständig bestimmter Vektor. Sobald die Relativgeschwindigkeit ermittelt ist, läßt sich auch der Volumenstrom angeben.

$$v_D = \omega R \quad \text{mit} \quad R = \frac{d}{2} \tag{A.6}$$

$$v_a^2 = v_r^2 + v_D^2 - 2v_r \cos \alpha \cos \beta \cdot v_D \quad (\text{A.7})$$

$$v_r = v_D \cos(\alpha) \cos(\beta) + \sqrt{v_a^2 - v_D^2 (\cos^2(\alpha) \cos^2(\beta) - 1)} \quad (\text{A.8})$$

$$\dot{Q} = 2v_r A_q \quad (\text{A.9})$$

$$\mathbf{v}_r = \begin{bmatrix} v_{rt} \\ v_{rr} \\ v_{rv} \end{bmatrix} = v_r \cdot \begin{bmatrix} \cos(\alpha) \cos(\beta) \\ \cos(\alpha) \sin(\beta) \\ \sin(\alpha) \end{bmatrix} \quad (\text{A.10})$$

$$\mathbf{v}_a = \begin{bmatrix} v_{at} \\ v_{ar} \\ v_{av} \end{bmatrix} = \mathbf{v}_r - \begin{bmatrix} \omega \cdot R \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.11})$$

Damit sind alle Größen der Momentengleichungen bis auf ω verfügbar. ω ergibt sich iterativ aus der Forderung nach einem Gleichgewicht der Drehmoment. Ausgehend von einem konservativ abgeschätzten Startwert für ω fährt der Rasensprenger in der Simulation an und erreicht beim Momentengleichgewicht eine Grenzdrehzahl. Die Reibungsenergie wird dem System entzogen und äußert sich durch einen Druckverlust, da es außer dem Wasserstrahl keine weitere Energiequelle gibt.

$$\Delta p_v = \frac{M_R \omega}{\dot{Q}} \quad (\text{A.12})$$

Hinzu kommt der Druckverlust der querschnittsarmen Zuleitung zur Düse innerhalb des Rasensprengers. Nur bei großen Volumenströmen in Verbindung mit einem engen Zuleitungsquerschnitt spielt dies eine Rolle. Näherungslösungen sind also an dieser Stelle unkritisch. Aus tabellierten Daten folgt für glatte Rohre mit der Länge 300mm mit einem Durchmesser von 5mm bis 10mm und einem Volumenstrom zwischen 1 l/min und 10 l/min eine kompakte Näherung für den Verlustkoeffizienten.

$$\Delta p_{zul} = c_v \dot{Q}^2 \quad (\text{A.13})$$

$$c_v = 60000^2 \cdot 10^{5,0704 - 0,579413d_{zul} + 0,196432d_{zul}^2} \quad (\text{A.14})$$

Der Vorfaktor ergibt sich durch die Umrechnung von $\frac{l}{min}$ in $\frac{m^3}{s}$, da die Berechnung im Gegensatz zu den tabellierten Daten durchgängig SI-Einheiten benutzt. Erst wenn ω gegen einen stabilen Wert konvergiert, erreichen auch die Druckverluste ihren Endwert. Die Simulation beginnt daher mit einem geeigneten Startwert für v_a , der korrigiert wird, sobald für die Druckverluste bessere Werte vorliegen.

Die Flugbahn der Wassertropfchen wäre eine Wurfparabel, wenn es keinen Luftwiderstand gäbe. Dieser ist jedoch nicht vernachlässigbar und bremst die Tropfchen in Abhängigkeit von Ihrer Größe und Fluggeschwindigkeit. Vereinfachend wurde

eine bewährte Formulierung von SCHADE, KUNZ übernommen [161], die auf Arbeiten von ABRAHAMS basiert und für einen weiten Bereich der Reynoldszahl gilt.

$$Re = \frac{v_a d_{Tr} \rho_L}{\eta_L} \quad (\text{A.15})$$

$$\zeta = \frac{24}{Re} \cdot \left(1 + 0,11\sqrt{Re}\right)^2 \quad (\text{A.16})$$

Als Konstanten wurden folgende Werte angesetzt:

$$\rho_L = 1,25 \frac{\text{kg}}{\text{m}^3} \quad (\text{A.17})$$

$$\eta_L = 1,82 \cdot 10^{-5} \frac{\text{kg}}{\text{ms}} \quad (\text{A.18})$$

$$\rho_W = 1000 \frac{\text{kg}}{\text{m}^3} \quad (\text{A.19})$$

Tropfendurchmesser und Tropfenmasse richten sich bei kleinen Düsenquerschnitten nach der Größe der Austrittsöffnung.

$$d_{Tr} = \sqrt{\frac{4A_q}{\pi}} \quad (\text{A.20})$$

$$m_{Tr} = \frac{\pi}{6} d_{Tr}^3 \rho_L \quad (\text{A.21})$$

Aus dem Luftwiderstandsbeiwert ergeben sich die Bremskraft und die Tropfenverzögerung. Die Richtung der Tropfenverzögerung hängt von der momentanen Tropfengeschwindigkeit ab und wird in jedem Zeitschritt neu ermittelt. Die Berechnung der Flugbahn ist abgeschlossen, wenn ihre vertikale Komponente den Wert Null erreicht hat.

$$F_W = \frac{\rho_L}{2} v_a^2 A_q \zeta \quad \text{mit} \quad a = \frac{F_W}{m_{Tr}} \quad (\text{A.22})$$

$$s_{h0} = 0 \quad s_{v0} = 1 \text{ mm} \quad (\text{A.23})$$

$$v_{h0} = v_a \cos(\alpha) \quad v_{v0} = v_a \sin(\alpha) \quad (\text{A.24})$$

$$a_h = a \cos(\varphi) \quad a_v = a \sin(\varphi) \quad (\text{A.25})$$

$$\cos(\varphi) = \frac{v_h(t)}{v(t)} \quad \sin(\varphi) = \frac{v_v(t)}{v(t)} \quad (\text{A.26})$$

$$v_h = v_{h0} + \int_0^t a_h dt \quad v_v = v_{v0} + \int_0^t a_v dt \quad (\text{A.27})$$

$$s_h = s_{h0} + \int_0^t v_h dt \quad s_v = s_{v0} + \int_0^t v_v dt \quad (\text{A.28})$$

A.3 Erweiterungen

Basierend auf der dargestellten Basisvariante des Berechnungsprogramms wurden verschiedene optionale Erweiterungen eingeführt, um größere Stufenbreiten rechnen zu können. Dadurch entstehen stark nichtlineare Zusammenhänge zwischen den Eingangsgrößen und den Qualitätsmerkmalen. Für die Verdeutlichung der aufwendigen Verfahren (Kriging, Radial Based Functions, neuronale Netze etc.) war dies erforderlich.

Druckverlust

Die Berechnung des Druckverlustes in der Zuführleitung kann neben der Basisvariante ebenfalls durch den in Gleichung A.29 dargestellten Ansatz ermittelt werden. Durch die separate Berechnung des Reibungskoeffizienten λ können neben laminaren ebenfalls turbulente Strömungsverluste berücksichtigt werden, worauf im Rahmen dieser Arbeit verzichtet wird. Weiterhin wird im Gegensatz zur Basisvariante nicht ein Zulauf *vor* dem Rasensprenger angenommen, sondern die Arme selbst als Zulauf betrachtet, so dass neben dem Durchmesser d_{zul} der Leitung ebenfalls der Radius R des Rasensprengers einen direkten Einfluss auf den Druckverlust aufweist.

$$\Delta p_{zul} = \lambda \frac{\rho}{2} \frac{R}{d_{zul}} v_{zul}^2 \quad (\text{A.29})$$

mit $v_{zul} = \frac{\dot{Q}}{A_{zul}}$ und $\lambda_{laminar} = \frac{64}{Re} = 64 \frac{\nu}{d_{zul} v_{zul}}$

Flugweite

Soll der Radius R des Rasensprengers in der Flugweitenbestimmung berücksichtigt werden, so kann die Flugweite s^* mit der Basisflugweite s_h bestimmt werden.

$$s^* = \sqrt{[R + \sin(\beta) s_h]^2 + [\cos(\beta) s_h]^2} \quad (\text{A.30})$$

Haftreibung

Werden die Faktoren α , β und M_{Rt} in großen Bereichen variiert, so treten Faktorkombinationen auf, bei denen das Antriebsmoment des Wasserstrahls geringer ist als das Reibmoment M_{Rt} , so dass keine Rotation des Rasensprengers auftritt ($n = 0$). Da der implementierte Lösungsalgorithmus in diesen speziellen Fällen keinen Gleichgewichtszustand findet, wird vor der Volumenstromberechnung geprüft, ob die Haftreibung bei $n = 0$ überwunden wird. Ist dieses nicht der Fall, so wird die Flugweite mit dem berechneten Volumenstrom \dot{Q}_0 und der Drehzahl $n = 0$ ermittelt.

$$M_A = \dot{Q}_0 \rho R v_t \quad (\text{A.31})$$

$$v_t = \frac{\dot{Q}_0}{2A_q \cos(\alpha) \cos(\beta)}$$

$$\text{Energieerhaltung: } p_{in} - \Delta p_{zul} = \frac{\rho}{2} v_{aus}^2 \quad (\text{A.32})$$

$$\begin{aligned}
 \text{Basisansatz : } \Delta p_{zul} &= c_v \dot{Q}_0^2 \\
 \Rightarrow p_{in} - c_v \dot{Q}_0^2 &= \frac{\rho}{2} v_{aus}^2 = \frac{\rho}{2} \left(\frac{\dot{Q}_0}{2A_q} \right)^2 \\
 \Rightarrow \dot{Q}_0 &= \sqrt{\frac{p_{in}}{c_v + \frac{\rho}{8A_q^2}}}
 \end{aligned} \tag{A.33}$$

$$\begin{aligned}
 \text{Erweiterter Ansatz : } \Delta p_{zul} &= 64 \frac{v}{d_{zul} v_{zul}} \frac{\rho}{2} \frac{R}{d_{zul}} v_{zul}^2 \\
 \Rightarrow p_{in} - 64 \frac{v}{d_{zul} v_{zul}} \frac{\rho}{2} \frac{R}{d_{zul}} v_{zul}^2 &= \frac{\rho}{2} \left(\frac{\dot{Q}_0}{2A_q} \right)^2 \\
 \Rightarrow p_{in} - 64 \frac{v}{2d_{zul}^2} \rho R \frac{\dot{Q}_0}{A_{zul}} &= \frac{\rho}{2} \left(\frac{\dot{Q}_0}{2A_q} \right)^2 \\
 \Rightarrow \frac{2}{\rho} p_{in} - 64 \frac{v}{2d_{zul}^2 A_{zul}} R \dot{Q}_0 &= \left(\frac{\dot{Q}_0}{2A_q} \right)^2 \\
 \Rightarrow \dot{Q}_0^2 + \frac{32RA_q^2 v}{d_{zul}^2 A_{zul}} \dot{Q}_0 - \frac{2A_q^4}{\rho} p_{in} &= 0 \\
 \text{mit } c_{zul} &= \frac{128RA_q^2 v}{d_{zul}^2 A_{zul}} \\
 \Rightarrow \dot{Q}_0 &= -\frac{c_{zul}}{2} + \sqrt{\left(\frac{c_{zul}}{2} \right)^2 + \frac{8A_q^4}{\rho} p_{in}}
 \end{aligned} \tag{A.34}$$

Kodierung

Neben der Standardkodierung $[-1; 1]$ sind insgesamt folgende Faktorkodierungen implementiert worden:

- keine Kodierung
- $[-1; 1]$
- $[0; 1]$
- $[1, 2, 3, \dots, n_s]$

Variablen-Übergabe

Zur Erleichterung einer Automatisierung verschiedener Versuchsläufe können der Berechnungsfunktion folgende Variablen übergeben werden:

- *FileName*: Name der Eingabedatei (z.B. '1128.inp')
- *Fak*: Faktoreinstellungen [Spalte der Eingabedatei, Min, Max]

Ein Beispiel für eine mögliche Faktoreinstellung ist:

```

Fak=[1, 0 , 90 % alpha [°]
     2, 0 , 90 % beta [°]
     3, 2e-6, 4e-6 % Aquer [mm^2]
     4, 0.1 , 0.2 % Durchmesser [m]
     5, 0.01, 0.02 % Mtrocken [Nm]
     6, 0.01, 0.02 % Mfluessig [Nm/s]
     7, 1 , 2 % Druck [bar]
     8, 7 , 10 ]; % Durchmesser Zuleitung [mm]

```

A.4 Quellcode

Der folgende Quellcode ist unter Octave und Matlab lauffähig. Erforderlich ist eine Eingabedatei mit dem gewünschten Versuchsplan passend zu der gewählten Kodierung (zum Beispiel -1 bis 1). Jede Zeile entspricht einer separaten Berechnung mit der entsprechenden Faktoreneinstellung. Der gesamte Versuchsplan wird per Stapelverarbeitung mit einem Aufruf abgearbeitet.

```

function Rasensprenger (FileName,Fak)
% Rasensprengerversuch
% (c) 2002 - 2009
% Hauptteil: Karl Siebertz
% Erweiterung: David van Bebber
%
% Grundlegende Einstellungen
Kodierung      = 1; % 0Keine 1[-1;+1] 2[0;+1] 3[1,2,...,ns]
dpzulVariante = 1; %      1Basis  2Variation
sflVariante   = 1; %      1Basis  2Variation
% Konstanten
g=10; pi=3.141592654; rho=1000; dynVis=1;
kinVis=dynVis/rho; MaxFehler=0.005;
% Datei mit Parameterbelegung einlesen
if nargin < 1
    FileName = 'l128.inp';
end
s=load(FileName);
[kzei, kspa] = size(s);
if nargin >= 2
% Umrechnungsfaktoren für kodierte Daten
PF = 1; % Position des Faktors
[aspa, amin, aplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[bspa, bmin, bplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[cspa, cmin, cplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[dspa, dmin, dplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[espa, emin, eplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[fspa, fmin, fplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[gspace, gmin, gplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
[hspa, hmin, hplu]=SetD(Fak(PF,1),Fak(PF,2),Fak(PF,3));PF=PF+1;
else
[aspa, amin, aplu]=SetD(1, 0 , 90 ); % alpha [°]
[bspa, bmin, bplu]=SetD(2, 0 , 90 ); % beta [°]
[cspace, cmin, cplu]=SetD(3, 2e-6, 4e-6); % Aquer [mm^2]
[dspace, dmin, dplu]=SetD(4, 0.1 , 0.2 ); % Durchmesser [m]
[espace, emin, eplu]=SetD(5, 0.01, 0.02); % Mtrocken [Nm]
[fspace, fmin, fplu]=SetD(6, 0.01, 0.02); % Mfluessig [Nm/s]
[gspace, gmin, gplu]=SetD(7, 1 , 2 ); % Druck [bar]
[hspace, hmin, hplu]=SetD(8, 7 , 8 ); % Durchm. Zuleitung [mm]
end
% Ausgabedateien
ido0=fopen('d-kompl.dat','w');ido1=fopen('d-qm1.dat','w');
ido2=fopen('d-qm2.dat','w');ido3=fopen('d-qm3.dat','w');

```



```

ido4=fopen('d-qm.dat' , 'w');ido5=fopen('d-par.dat','w');
% große Schleife
ominc=0;
for j=1:kzei
    n=0;sfl=0;qp=0;
% Auswahl der Kodierungsfunktion
    switch Kodierung
        case 0
            NormFunc = @Norm0;
        case 1
            NormFunc = @Norm1;
        case 2
            NormFunc = @Norm2;
        case 3
            NormFunc = @Norm3;
        otherwise
            break
    end
% Berechnung der Parameter
    alpha= feval(NormFunc,amin,aplu,s,j,aspa);
    beta = feval(NormFunc,bmin,bplu,s,j,bspa);
    A = feval(NormFunc,cmin,cplu,s,j,cspa);
    d = feval(NormFunc,dmin,dplu,s,j,dspa);
    mt = feval(NormFunc,emin,eplu,s,j,espa);
    mf = feval(NormFunc,fmin,fplu,s,j,fspace);
    pin = feval(NormFunc,gmin,gplu,s,j,gspa);
    dzul = feval(NormFunc,hmin,hplu,s,j,hspa);
    pin = pin * 1e5;
    h = pin * 1e-4;
    R = d/2;
    sina = sin(alpha*pi/180); cosa = cos(alpha*pi/180);
    tana = tan(alpha*pi/180); sinb = sin(beta*pi/180);
    cosb = cos(beta *pi/180); tanb = tan(beta*pi/180);
% Interpolation gültig für d von 5mm bis 10mm
    cvzul = 10^(5.0704 -0.579413*dzul+0.0196432*dzul^2);
    dzul = (cvzul*60000^2);
% Startwerte
    m0 = 2*rho*A*R*2*g*h*cosa+cosb;
    n1 = 0.1*abs(m0-mt)/(mf+5.0e-4);
    omega = 2*pi*n1;
    msoll = mt+omega*mf;
    mdiff = m0;
    va0 = sqrt(2*pin/rho);
    deltap= abs(msoll*omega)/(A*va0); % Verlustleistung Startwert
    dzul = dzul * 1e-3;
    Azul = pi/4*dzul^2;
    if dpzulVariante == 1
        dpzul=cvzul*(A*va0)^2;
    else
        dpzul=0.1*pin;
    end
    it=0; va=0; vr=0; m=m0;
% Durchfluss bei n = 0
    if dpzulVariante == 1

```

```

qp= sqrt (pin/ (cvzul+rho/8/A^2));
else
  c = 128*R*A^2*kinVis/ (dzul^2*Azul);
  qp= -c/2+sqrt ((c/2)^2+8/rho*pin*A^2);
end
va = qp/2/A;
vr = va;
vrt= va*cosb*cosa;
vat= vrt;
m = rho*qp*R*vat;
if m>mt % Haftreibung überschritten?
% Iteration bis zum Momentengleichgewicht
  while abs(mdifff) > MaxFehler*abs(m)
    n = omega/2/pi;
    msoll= mt+n*mf;
    varm = omega*R;
    % Energiebilanz des gesamten Rasensprengers
    pen = pin-deltap-dpzul;
    if (pen < 0.01*pin)
      fprintf('Fehler: pen < 0.01*pin\n');
      pin,deltap,dpzul,msoll
      m0,m,mdifff,ominc,vr,varm,va
      va=0;
      break;
    end
    va=sqrt (2*pen/rho);
    if (va^2+varm^2*(cosa^2*cosb^2-1) < 0 )
      fprintf('Fehler: va^2+varm^2*(cosa^2*cosb^2-1)<0\n');
      va,vr,varm,vak
      break;
    end
    vr = varm*cosa*cosb;
    vr = vr+sqrt (va^2+varm^2*(cosa^2*cosb^2-1));
    vrt = vr*cosb*cosa;
    vrr = vr*cosa*sinb;
    vrv = vr*sina;
    vat = vrt-omega*R;
    var = vrr;
    vav = vrv;
    % Kontrolle der Komponentenzerlegung
    vak = sqrt (vat^2+var^2+vav^2);
    m = 2*rho*vr*A*R*vat;
    mdifff = m-msoll;
    ominc = 0.1*min(abs(mdifff/m), (0.5*pen/pin));
    % variable Schrittweite
    omega = omega*(1+ominc)^sign(mdifff);
    qp = 2*vr*A;
    % Verlustleistung in Druck umgerechnet
    deltap= abs(msoll*omega)/qp;
    if dpzulVariante == 1
      dpzul= cvzul*qp^2;
    else
      vzul = qp/2/Azul; % qp/2 durch einen Arm
      Re = abs(dzul*vzul/kinVis);
    end
  end
end

```

```

        dpzul= 64/Re*rho/2*R/dzul*vzul^2;
    end
    it=it+1;
    if it > 10000
        fprintf('Fehler: it > 10000\n');
        it,msoll,mdiff,ominc,alpha,beta
        A,d,mt,mf,vr,va,vrt,varm,vat,omega
        break;
    end
    if(omega < 0.0062 )
        fprintf('Fehler: omega < 0.0062\n');
        it,omega
        n=0;
        break;
    end
end
else
    omega=0; n=0;
end
% Flugbahn
dtropf = sqrt(4*A/pi);
etaluft= 1.82e-5;
nyluft = etaluft/1.25;
v      = va;
z      = 1.0e-3;
sfl    = 0.0;
vh     = va*cosa;
vv     = va*sina;
deltat = 0.005;
mtr    = pi/6*dtropf^3*rho;
while z > 0
    if(va<0.01)
        break;
    end
    Re = va*dtropf/nyluft;
    % Abraham, The Physics of Fluids 13, S.2194
    zeta= 24/Re*(1+0.11*sqrt(Re))^2;
    Fwid= 1.25/2*va^2*pi/4*dtropf^2*zeta;
    atr = Fwid/mtr;
    sfl = sfl+vh*deltat;
    z   = z+vv*deltat;
    vh  = vh-atr*cosa*deltat;
    vv  = vv-g*deltat-atr*sina*deltat;
    va  = sqrt(vh^2+vv^2);
    cosa= vh/va;
    sina= vv/va;
end;
if sflVariante == 1
    sfl=sfl;
else
    sfl=sqrt((R+sinb*sfl)^2+(cosb*sfl)^2);
end
qp      = 2*vr*A*60000;
pverh= deltap/(rho*g*h);

```

```

% Ausgabe
Amm2 = 1000000*A;
dmm  = 1000*d;
mtmm = mt*1000;
mfmm = mf*1000;
dzulmm= dzul*1e3;
fprintf(ido0,'%6.2e %6.2e %6.2e %6.2e ',alpha,beta,Amm2,dmm);
fprintf(ido0,'%6.2e %6.2e %6.2e %6.2e ',mtmm,mfmm,h,dzulmm);
fprintf(ido0,'%10.8e %10.8e %10.8e \n',n,sfl,qp);
fprintf(ido1,'%10.4f \n',n);
fprintf(ido2,'%10.4f \n',sfl);
fprintf(ido3,'%10.8f \n',qp);
fprintf(ido4,'%10.8f %10.8f %10.8f \n',n,sfl,qp);
fprintf(ido5,'%6.2e %6.2e %6.2e %6.2e ',alpha,beta,Amm2,dmm);
fprintf(ido5,'%6.2e %6.2e %6.2e %6.2e \n',mtmm,mfmm,h,dzulmm);

end;
fclose(ido0);fclose(ido1);fclose(ido2);
fclose(ido3);fclose(ido4);fclose(ido5);
% Hilfsfunktionen
function Value=Norm0(MinVal,MaxVal,data,row,col) % ohne Kodierung
if size(data,2) < col || size(data,1) < row
    Value = (MinVal+MaxVal)/2;
else
    Value = data(row,col);
end
function Value=Norm1(MinVal,MaxVal,data,row,col) % [-1;1]
if size(data,2) < col || size(data,1) < row
    Value = (MinVal+MaxVal)/2;
else
    Value = MinVal+(MaxVal-MinVal)*(data(row,col)+1)/2;
end
function Value=Norm2(MinVal,MaxVal,data,row,col) % [0;1]
if size(data,2) < col || size(data,1) < row
    Value = (MinVal+MaxVal)/2;
else
    Value = MinVal+(MaxVal-MinVal)*data(row,col);
end
function Value=Norm3(MinVal,MaxVal,data,row,col) % [1,2,...,ns]
if size(data,2) < col || size(data,1) < row
    Value = (MinVal+MaxVal)/2;
else
    minStufe=min(data(:,col));data(:,col)=data(:,col)-minStufe+1;
    maxStufe=max(data(:,col));diffStufe=maxStufe-1;
    if diffStufe == 0
        Value=(MinVal+MaxVal)/2;
    else
        Value=MinVal+(MaxVal-MinVal)*(data(row,col)-1)/diffStufe;
    end
end
function [spalte,minus,plus]=SetD(Spalte,Minimal,Maximal)
spalte=Spalte;minus=Minimal;plus=Maximal; % Set Factor Data

```