Electronic Supplementary Material (ESI) for Soft Matter.

Supplementary Information:

Dynamics of fog droplets on a harp wire

Nicholas G. Kowalski and Jonathan B. Boreyko*

Department of Mechanical Engineering, Virginia Tech, Blacksburg, Virginia 24061, United States.

E-mail: boreyko@vt.edu

Experimental setup

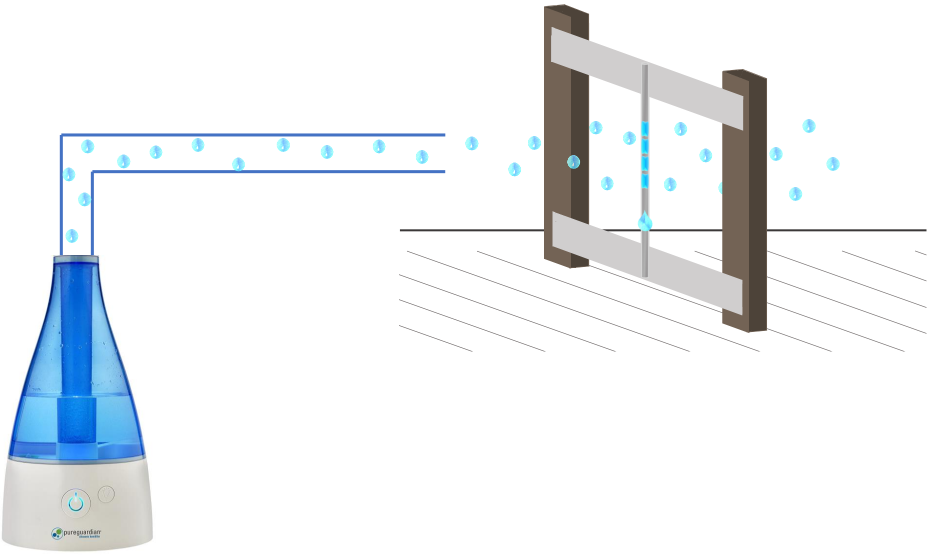


Fig. S1: Schematic of the experimental setup. An ultrasonic humidifier was connected to a directional tube to provide a consistent fog stream. A single wire was strung in the center of a frame and placed at the center of the incoming fog stream.

Numerical loop

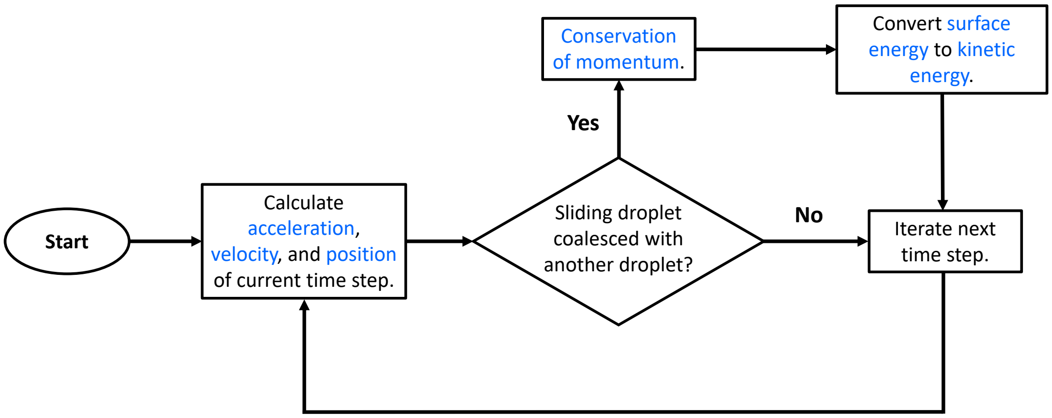


Fig. S2: Basic visualization of the numerical loop used to calculate the acceleration, velocity, and position of the main sliding droplet at the end of each time step. All three values are calculated at the beginning of the loop. Afterwards, the code checks to see if the sliding droplet has slid a sufficient distance since its last coalescence event to meet another static droplet along the wire (controlled primarily by b_z). If it has, the code will retroactively account for change in velocity from both conservation of momentum and the change in combined surface energy.

Dominant forces in sliding dynamics

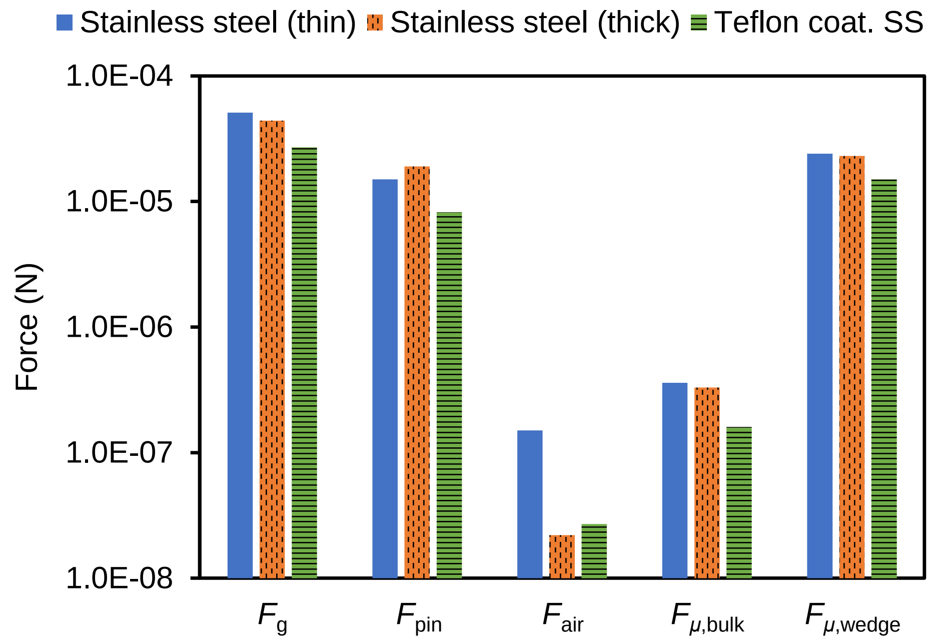


Fig. S3: Logarithmic plot showcasing the magnitude of forces acting on the sliding droplet for stainless steel wire of $r_{\mathrm{w}} = 0.127\,\mathrm{mm}$ (blue solid bars), stainless steel wire of $r_{\mathrm{w}} = 0.254\,\mathrm{mm}$ (orange vertical-dash bar), and Teflon-coated wire of $r_{\mathrm{w}} = 0.127\,\mathrm{mm}$ (green horizontal-line bar). The average of all trials for each wire type is plotted for a given force. Each value is from the last iteration of each respective simulation. It can be seen that F_{air} and $F_{\mu,\mathrm{bulk}}$ are 1–3 orders of magnitude smaller than F_{g}, F_{pin}, and $F_{\mu,\mathrm{wedge}}$, causing the latter three to be the dominant forces for sliding droplet dynamics.

Simulations without $F_{\mu,\mathbf{wedge}}$

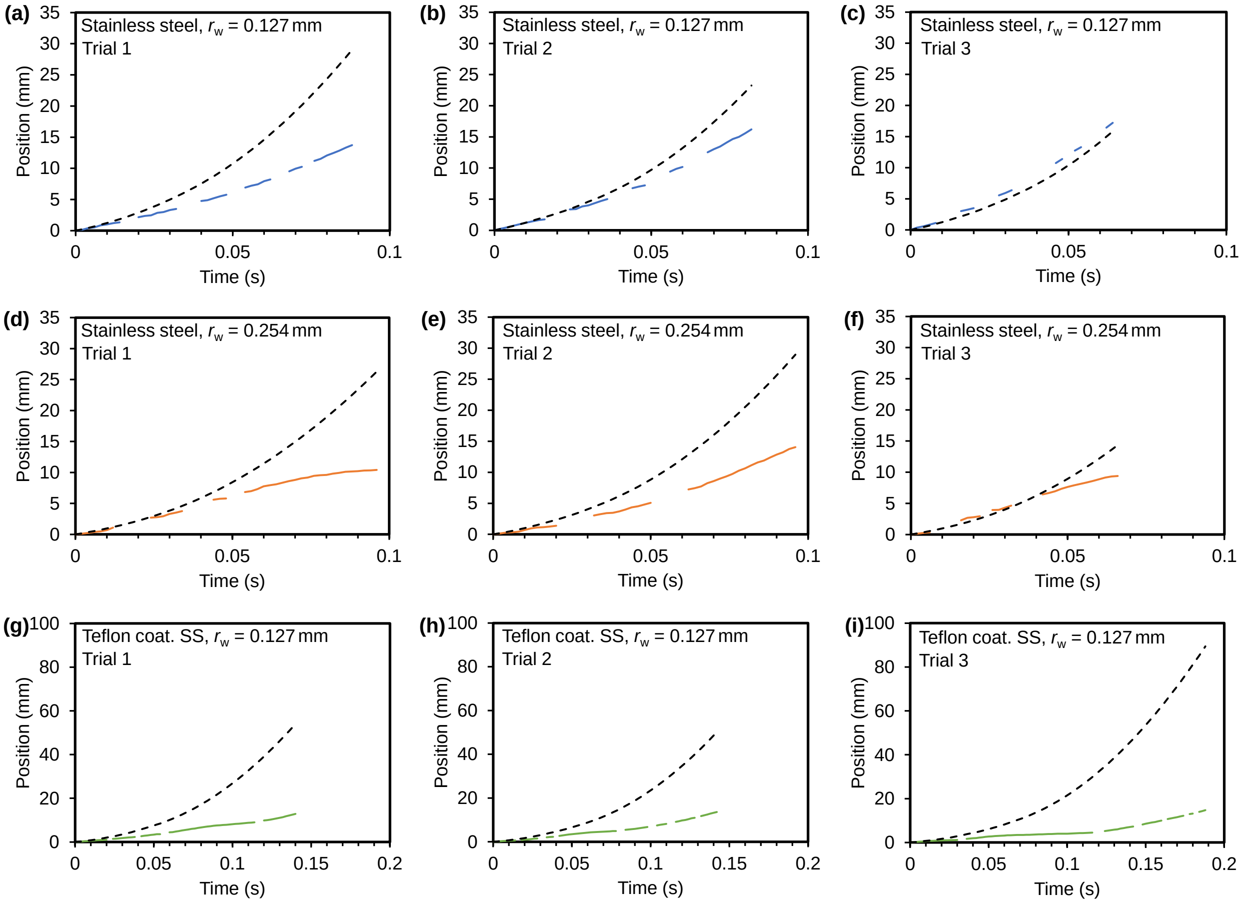


Fig. S4: Comparison of the experimental (solid lines) and theoretical (dashed lines) displacements of fog droplets sliding down a vertical harp wire. In contrast to Fig. 4 of the main manuscript, the simulations used to generate the theoretical data here did not include $F_{\mu,\text{wedge}}$ as a resisting force. The result of this exclusion is simulated droplets that generally accelerate much faster than if $F_{\mu,\text{wedge}}$ were present, particularly on the Teflon-coated wire. This highlights the need for including $F_{\mu,\text{wedge}}$ as a resisting force, as something is clearly amiss if we only consider $F_{\mu,\text{pin}}$, F_{air}, and $F_{\mu,\text{bulk}}$ as resisting forces. The inclusion of $F_{\mu,\text{wedge}}$ in our analysis, along with its noted presence on droplets in our experiments, helps to fill this gap.

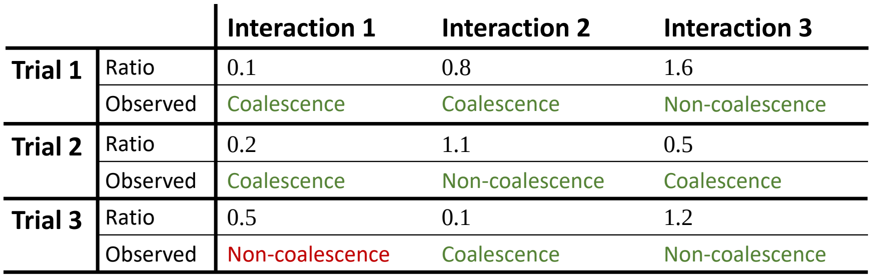
Input data

Table S1: Table of properties used to generate the theoretical data in Fig. 4 of the main manuscript for each wire type. θ_r and θ_a represent the average of five measurements of different critical droplets (± 2 standard deviations). S_1 and S_2 are constants obtained from a power-law fit of measured $2r_{\mathrm{M},j}$ versus q_j values (± 2 standard deviations).

	θ_r (deg)	θ_a (deg)	S_1	S_2	β
Stainless steel, r_w = 0.127 mm	43 ± 12	77 ± 4	3 ± 2	0.2 ± 0.1	0.08
Stainless steel, r_w = 0.254 mm	45 ± 2	68 ± 5	2 ± 1	0.23 ± 0.07	0.05
Teflon coat. SS, r_w = 0.127 mm	106 ± 8	124 ± 14	0.5 ± 2	-0.09 ± 0.05	0.03

Non-coalescence model

Table S2: Table of experimental $F_\mathrm{c}/F_\mathrm{pin}$ ratios for the Teflon-coated wire trials calculated using eqn (14) of the main manuscript. For early times, a value of $F_\mathrm{c}/F_\mathrm{pin} < 1$ predicts coalescence, while a value > 1 predicts non-coalescence. The measurements taken to use in eqn (14) were right before the first three collisions of the sliding droplet with static droplets for each trial. The observed outcome of each interaction is noted below each calculated ratio to highlight whether it agrees (green) or disagrees (red) with the predicted behavior. Of the nine ratios calculated, only one (Interaction 1 of Trial 3) disagreed with experimental observation.



		Interaction 1	**Interaction 2**	**Interaction 3**
Trial 1	Ratio	0.1	0.8	1.6
	Observed	Coalescence	Coalescence	Non-coalescence
Trial 2	Ratio	0.2	1.1	0.5
	Observed	Coalescence	Non-coalescence	Coalescence
Trial 3	Ratio	0.5	0.1	1.2
	Observed	Non-coalescence	Coalescence	Non-coalescence

Viscous dissipation at terminal velocity

Table S3: Table of viscous dissipation values at terminal velocity. All values correlate with the simulated trials shown in Fig. 4 of the main manuscript ($F_{\mu,\text{wedge},1}$ for stainless steel wire of $r_\text{w} = 0.127\,\text{mm}$ corresponds with Fig. 4a, $F_{\mu,\text{wedge},2}$ for stainless steel wire of $r_\text{w} = 0.254\,\text{mm}$ corresponds with Fig. 4e, etc). The simulated trials were all run identically to their respective counterparts in Fig. 4, but were extended well beyond the final coalescence event to reach terminal velocity (i.e. to where $a \approx 0$). Results show that, while $F_{\mu,\text{wedge}}$ is larger than F_pin in almost all cases, the latter can certainly not be neglected. The only wire type where $F_{\mu,\text{wedge}}$ is considerably larger than F_pin by the time terminal velocity is achieved is the Teflon-coated stainless steel; in all other cases, the two forces are of the same order of magnitude.

	F_pin (N)	$F_{\mu,\text{wedge},1}$ (N)	$F_{\mu,\text{wedge},2}$ (N)	$F_{\mu,\text{wedge},3}$ (N)
Stainless steel, r_w = 0.127 mm	1.5e-05	3.2e-05	3.0e-05	4.3e-05
Stainless steel, r_w = 0.254 mm	1.9e-05	1.9e-05	2.3e-05	2.1e-05
Teflon coat. SS, r_w = 0.127 mm	8.2e-06	1.7e-05	1.7e-05	2.1e-05

MATLAB code

```
clear;
clc;
close all

% By Nick Kowalski
% NIFI lab, Virginia Tech
% Fog harvesting project

%% Introduction
disp('> The following program is paired with the manuscript "Dynamics of fog droplets on a harp wire" by');
disp('  Nicholas Kowalski and Jonathan Boreyko.  The program will calculate position, velocity, and');
disp('  acceleration of a water droplet sliding down a vertical wire, merging with other droplets along');
disp('  the way.  The model system considers a vertical wire with n droplets along it, where each droplet');
disp('  is initially static.  Each droplet is slowly growing over time due to an incoming fog stream.');
disp('  Appreciable droplet sliding initiates by the coalescence of a top pair of droplets; the merged');
disp('  droplet subsequently coalesces with all underlying droplets as it moves down the wire.');
disp('> When altering the physical system, please refer to line 183 of the code to make appropriate parameter');
disp('  changes.');
disp('> Refer to Figure 1 in the main manuscript for a visual representation of the system as needed.');
disp('---');

%% Inputs
k = 1;
while k == 1
    z = input('> Would you like to approximate all droplets as perfect spheres (1 for yes, 0 for no): ');
    if z == 1
        k = 0;
    elseif z == 0
        k = 0;
    else
        disp('  Invalid input; must enter 1 or 0.');
    end
end

k = 1;
while k == 1
    NumD = input('> Enter the number of droplets along the wire (including the intial coalescing pair): ');
    if NumD < 2
        disp('  Invalid input; number must be greater than or equal to two.');
```

```
    else
        k = 0;
    end
end

k = 1;
while k == 1
    DataI = input('> Import data from a spreadsheet, or enter manually (1 for former, 0 for latter): ');
    if DataI == 1
        k = 0;
    elseif DataI == 0
        k = 0;
    else
        disp('  Invalid input; must enter 1 or 0.');
    end
end

if DataI == 1
    disp('> Ensure the file to read is named "data.txt" in the same file location as the MATLAB code.');
    disp('  Values should be arranged in a single column, starting with the "2r_(M,j)" values, directly');
    disp('  followed by "b_z" values.');
    disp('  Values should be in descending order from the top of the wire, in units of millimeters.');
    disp('  For the measured length "2r_(M,j)" values of droplets, start with the initial coalescing pair.');
    disp('  For spacing "b_z" values between pairs of droplets, start with the pair beneath the top');
    disp('  initial coalescing droplet.');
    disp('  Press Enter to continue.');

 k = 1;
    while k == 1
        flag = 0;
        input('');
        Imp = importdata('data.txt');

        DataL = 2*NumD - 2;

        if length(Imp) > DataL
            flag = 2;
        end

        if length(Imp) < DataL
            flag = 3;
```

```
end

if flag == 0
    Lmm = zeros(1,NumD);
    for NumDC = 1:NumD
        Lmm(1,NumDC) = Imp(NumDC,1);
        if Imp(NumDC,1) <= 0
            flag = 1;
        end
    end
    L = Lmm/1000;

    bmm = zeros(1,(NumD - 2));
    for NumDC = 1:(NumD - 2)
        bmm(1,NumDC) = Imp((NumDC + NumD),1);
        if Imp((NumDC + NumD),1) <= 0
            flag = 1;
        end
    end
    b = bmm/1000;
end

if flag == 1
    disp('> Invalid input; one of the entered values is less than or equal to zero.');
    disp('  To try again with new data: press Enter to continue.');
    disp('  To change the inputted number of droplets along the wire: terminate the program');
    disp('  (Ctrl+C) and re-run.');
elseif flag == 2
    disp('> Invalid input; too many values in the imported data file.');
    fprintf('  Data file must have %3.0f entries (from number of droplets inputted\n',DataL);
    disp('  along the wire).');
    disp('  To try again with new data: press Enter to continue.');
    disp('  To change the inputted number of droplets along the wire: terminate the program');
    disp('  (Ctrl+C) and re-run.');
elseif flag == 3
    disp('> Invalid input; too few values in the imported data file.');
    fprintf('  Data file must have %3.0f entries (from number of droplets inputted\n',DataL);
    disp('  along the wire).');
    disp('  To try again with new data: press Enter to continue.');
    disp('  To change the inputted number of droplets along the wire: terminate the program');
    disp('  (Ctrl+C) and re-run.');
```

```
        else
            k = 0;
        end
    end

    if length(b) > 0
        It = 1;
    else
        It = 0;
    end

else

    NumDC = 1;
    disp('> Enter the measured length "2r_(M,j)" of all droplets along the wire (including the intial');
    disp('  coalescing pair).');
    disp('  These will be entered in the order of descending the wire, starting with the top initial');
    disp('  coalescing droplet (see Figure 1 in the main manuscript).')
    while NumDC <= NumD
        fprintf('  Droplet #%3.0f.\n',NumDC);
        k = 1;
        while k == 1
            Lmm(NumDC) = input('  Enter the measured length "2r_(M,j)" of the stated droplet (mm): ');
            if Lmm(NumDC) <= 0
                disp('  Invalid input; measured length must be greater than zero.');
            else
                k = 0;
            end
        end
        NumDC = NumDC + 1;
    end
    L = Lmm/1000;

    if NumD > 2
        It = 1;
        NumDC = 1;
        disp('> Enter the spacing "b_z" between all pairs of droplets (excluding the initial coalescing');
        disp('  pair.');
        disp('  These will be entered in the order of descending the wire, starting with the pair');
        disp('  beneath the top initial coalescing droplet (see Figure 1 in the main manuscript).')
        while NumDC <= (NumD - 2)
```

```
            fprintf('  Droplet pair #%3.0f.\n',NumDC);
            k = 1;
            while k == 1
                bmm(NumDC) = input('  Enter the spacing between the stated pair (mm): ');
                if bmm(NumDC) <= 0
                    disp('  Invalid input; spacing must be greater than zero.');
                else
                    k = 0;
                end
            end
            NumDC = NumDC + 1;
        end
    else
        It = 0;
        bmm = 0;
    end
    b = bmm/1000;
end

%% Variables
% Adjust the following system properties in lines 184-190 as needed.
rwmm = 0.127;  % mm; radius of wire
t = 0.0001;    % s; time step
the_r = 43;    % degrees; receding contact angle of a critical/sliding drop, measured experimentally
the_a = 77;    % degrees; advancing contact angle of a critical/sliding drop, measured experimentally
S1 = 2.73;     % pre-factor for power law fit (from experimental data)
S2 = 0.2103;   % pre-factor for power law fit (from experimental data)
beta = 0.08;   % percent (decimal) of surface energy converted to kinetic energy when two drops coalesce

rw = rwmm/1000;
l = pi*rw;
gam = 0.073;
g = 9.81;
rhow = 1000;
mu = 0.0010016;
rhoa = 1.1225;

if z == 0
    q = @(y) S1*(y^S2);
else
    q = @(y) 1;
```

```
end


for k = 1:NumD
    V(k) = (4/3)*(pi)*(q(L(k))^2)*((L(k)/2)^3);
end


%% Initial sliding velocity
% Coalescence of two drops is what initiates significant sliding;
% A non-zero velocity at time t = 0 appears due to a reduction in
% surface energy after the initial coalescence occurs.
rt = L(1)/2;
rb = L(2)/2;
Vd = V(1) + V(2);


if z == 0
rd1 = rt + (rb/9);
k = 1;
    while k == 1
        Vguess = (4/3)*pi*((S1*(2*rd1)^S2)^2)*(rd1^3);
        if abs((Vd - Vguess)/Vd) < 0.001
            k = 0;
        elseif (Vd - Vguess) > 0
            rd1 = rd1 + 1e-7;
        else
            rd1 = rd1 - 1e-7;
        end
    end
else
    rd1 = ((3/4)*(1/pi)*Vd)^(1/3);
end


Ld1 = 2*rd1;
rd2 = q(Ld1)*rd1;


ft = (2*q(L(1))^1.6) + (q(L(1))^3.2);
fb = (2*q(L(2))^1.6) + (q(L(2))^3.2);
ff = (2*q(Ld1)^1.6) + (q(Ld1)^3.2);


dS = 4*pi*gam*( (((ft/3)^(5/8))*rt^2) + (((fb/3)^(5/8))*rb^2)...
   - (((ff/3)^(5/8))*rd1^2) );
veli = ((2*beta*dS)/(rhow*Vd))^(1/2);
```

```
%% Initializations for loop
NumDC = 1;
CoM = (((L(1)/2)+(L(2)/2))*(rhow*V(1)))/((rhow*V(1)) + (rhow*V(2)));
if It == 1
    count = b(NumDC) + CoM - (Ld1/2) - (L(3)/2);
end

dif = cosd(the_r) - cosd(the_a);
%Vc = (l*gam*dif)/(rhow*g);

vel(1) = veli;
Aproj1 = pi*(rd2^2);
if q(Ld1) == 1
    phi = 1;
    phic = 1;
    phil = 1;
else
    req = ((3*4)*Vd*(1/pi))^(1/3);
    SAeq = 4*pi*(req^2);
    SAd = 4*pi*(((2*((rd1*rd2)^1.6)) + ((rd2*rd2)^1.6))/3)^(1/1.6);
    phi = SAeq/ SAd;
    Aeq = pi*(req^2);
    phic = Aeq/ Aproj1;
    Aproj2 = pi*rd1*rd2;
    phil = Aeq/ ((.5*SAd) - Aproj2);
end
x(1) = 0;
Fpin = l*gam*dif;

%% Acceleration loop (numerical calculations)
for i = 1:1999
    if vel(i) == 0
        a(i) = 0;
        vel(i+1) = 0;
        x(i+1) = x(i);
    else
        Re = (rhow*(2*rd2)*vel(i))/mu;
        Cd1 = (8/Re)*(1/sqrt(phil));
        Cd2 = (16/Re)*(1/sqrt(phi));
        Cd3 = (3/sqrt(Re))*(1/(phi^(3/4)));
```

```
    Cd4 = -log10(phi);
    Cd5 = -(-Cd4)^.2;
    Cd6 = 0.42*(10^(0.4*Cd5))*(1/phic);
    Cd = Cd1 + Cd2 + Cd3 + Cd6;
    alp = 4;
    Fgrav = rhow*Vd*g;
    Fda = .5*Cd*rhoa*Aproj1*(vel(i)^2);
    Fvb = (2*pi*rd1*rw)*mu*(vel(i)/(2*rd2));
    Fvw = alp*gam*pi*rw*(((10*mu*vel(i))/gam)^(2/3));
    a(i) = (Fgrav - Fpin - Fda - Fvb - Fvw)/ (rhow*Vd);
    vel(i+1) = vel(i) + a(i)*t;
    if vel(i+1) <= 0
        a(i) = 0;
        vel(i+1) = 0;
        x(i+1) = x(i);
    else
        x(i+1) = x(i) + vel(i)*t + .5*a(i)*(t^2);
    end
end

if It == 1
    j = 1;
else
    j = 0;
end
while j == 1
    if x(i+1) > count
        NumDC = NumDC + 1;
        NumDL = NumDC + 1;

        Ei = .5*rhow*Vd*(vel(i+1)^2);
        vel(i+1) = (Vd*vel(i+1))/(Vd + V(NumDL));
        Em = .5*rhow*(Vd + V(NumDL))*(vel(i+1)^2);
        dEm = Em - Ei;

        ri = rd1;
        Li = Ld1;
        Vi = Vd;
        Vd = Vd + V(NumDL);

        if z == 0
```

```
    rd1 = ri + ((L(NumDL)/2)/9);
    k = 1;
    while k == 1
        Vguess = (4/3)*pi*((S1*(2*rd1)^S2)^2)*(rd1^3);
        if abs((Vd - Vguess)/Vd) < 0.001
            k = 0;
        elseif (Vd - Vguess) > 0
            rd1 = rd1 + 1e-7;
        else
            rd1 = rd1 - 1e-7;
        end
    end
else
    rd1 = ((3/4)*(1/pi)*Vd)^(1/3);
end

Ld1 = 2*rd1;
rd2 = q(Ld1)*rd1;

fd = (2*q(Li)^1.6) + (q(Li)^3.2);
fb = (2*q(L(NumDL))^1.6) + (q(L(NumDL))^3.2);
ff = (2*q(Ld1)^1.6) + (q(Ld1)^3.2);

dS = 4*pi*gam*( (((fd/3)^(5/8))*ri^2) + (((fb/3)^(5/8))...
   *(L(NumDL)/2)^2) - (((ff/3)^(5/8))*rd1^2) );
Ef = Ei + dEm + beta*dS;
vel(i+1) = ((2*Ef)/(rhow*Vd))^(1/2);

Aproj1 = 2*pi*(((((rd1*rd2)^1.6) + (2*(rd2^3.2)))/3)^(1/1.6));
if q(Ld1) == 1
    phi = 1;
    phic = 1;
    phil = 1;
else
    req = ((3*4)*Vd*(1/pi))^(1/3);
    SAeq = 4*pi*(req^2);
    SAd = 4*pi*(((2*((rd1*rd2)^1.6))...
        + ((rd2*rd2)^1.6))/3)^(1/1.6);
    phi = SAeq/ SAd;
    Aeq = pi*(req^2);
    phic = Aeq/ Aproj1;
```

```
                Aproj2 = pi*rd1*rd2;
                phil = Aeq/ ((.5*SAd) - Aproj2);
            end

            if NumDC < (NumD - 1)
                CoM = (((Li/2)+(L(NumDL)/2))*(rhow*Vi))/((rhow*Vi)...
                    + (rhow*V(NumDL)));
                xCoM = (count + ri + (L(NumDL)/2)) - CoM;
                count = xCoM + CoM + b(NumDC) - rd1 - L(NumDL + 1);
            else
                It = 0;
                j = 0;
            end
        else
            j = 0;
        end
    end
end
x = x.*1000;
vel = vel.*1000;
a = a.*1000;
disp('---');
disp('> Program completed.');
fprintf('> Time step used was %6.5f seconds, beginning at t = 0 seconds.\n',t);
disp('See Workspace for position (x, mm), velocity (vel, mm/s), and acceleration (a, mm/s^2) at each time step.');
```

Supplementary movies

Movie S1: Playback of fog droplets sliding down a vertical stainless steel wire of radius 0.127 mm. The original footage was captured at 3,000 frames per second, played back here at a hundredth the original speed.

Movie S2: Playback of fog droplets sliding down a vertical stainless steel wire of radius 0.254 mm. The original footage was captured at 3,000 frames per second, played back here at a hundredth the original speed.

Movie S3: Playback of fog droplets sliding down a vertical Teflon-coated stainless steel wire of radius 0.127 mm. The original footage was captured at 3,000 frames per second, played back here at a hundredth the original speed.