2D Collisions of Stingers: A Study of Degration Over Time

10721 Research Immersion Course

By:
Shristhi Pant s25

Supervisors:
Kaar Hartvig Jenson
John Sebastian

July 25, 2025

Table of Contents

1 | Abstract

Nature's pointed structures, such as stingers and thorns, are often perfectly shaped for penetration and defense. Previous studies, like one by Quan et al. (2024), have focused on the geometry of these biological tools, showing that many follow a consistent taper-to-length ratio to maximize efficiency. However, those studies capture a single moment in time when natural tools degrade with use. This project explores how stinger-like structures change over time through repeated 2D mechanical collisions. Using colored pencils, chalk, and clay as stand-ins for biological materials, we analyzed wear patterns by tracking mass loss and tip sharpness through 2D image processing and MATLAB. Each material responded differently: chalk fractured quickly, colored pencils wore down gradually, and clay showed cracks. By introducing time as a variable, this study builds on existing geometric models and offers new insight into how materials and shape work together under stress.

2 | Introduction

Nature is armed with its own defense system, from the smallest of thorns of cacti and roses to the largest horns of rhinos and narwhal horns. These species of nature develop tapered pointed structures as an evolutionary tool for survival, it is used to defend and attack. In a study, "*The shape of Nature's stingers revealed"* (Quan et al. 2024) investigates these biological tools and their taper-to-length ratio to maximize penetration efficiency. They indicate that the tapering exponent will fall between 2 and 3 universally (Quan et al. 2024).

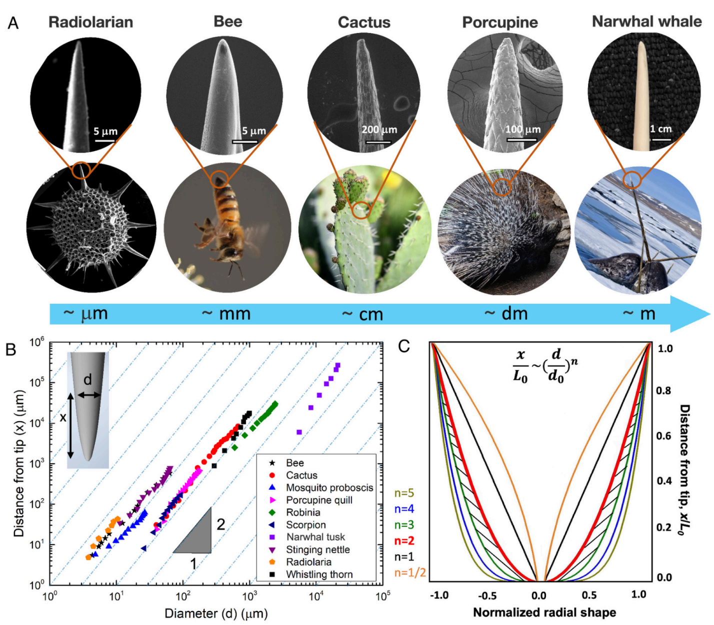


Figure 1: Quan

However, the paper examines characteristics of natural stingers at a single point in time and overlooks a key aspect of nature, time. The degradation of repeated use of a pointed stinger. Biological systems do not operate in a time-independent manner, friction, collision, and simply time all contribute to wear. Take shark teeth as an example, they are designed to kill prey, but through extended use, they lose their effectiveness over time, therefore needing to be replaced for optimal function.

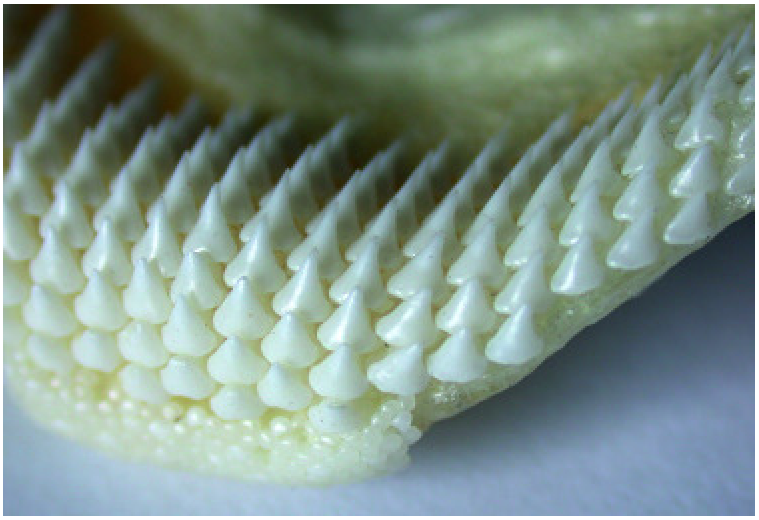


Figure 2: Shark Teeth Rows

This research builds on the geometric principles of Quan et al. while introducing a new variable, time. By 2D image analysis of stinger-like structures subjected to repeated mechanical collisions, this study investigates how shape and material influence degradation over time. We examine three materials with distinct wear properties: colored pencil, chalk, and clay. At each time step, we measure mass loss and sharpness reduction to assess how the structure evolves with repeated use. With this methodology, a question arises: If nature defines an optimal shape for stingers, how does repeated use affect the performance and shape of pointed objects over time? By introducing time as a variable, this study expands the understanding of tapered structures beyond ideal geometry to include the dynamics of wear. The results aim to contribute to broader applications in bioinspired design, where maintaining performance over time is as critical.

3 | Experimental Setup

This section presents the experimental setup and materials for executing a 2D motion analysis of stinger-like structures.

3.1 List of Items

- MATLAB
- Inverse Macroscope

- Raspberry Pie
- Scale
- Plate
- Plaster Vibration Machine
- Airdry Clay
- Clay Mold
- Colored Pencils
- Chalk
- File
- Foam
- Pencil sharpener

3.3 Setup

To begin the experiment, the colored pencils are sharpened until a point and smoothness is achieved. To determine sharpness, the pencil was analyzed under the inverse macroscope. When the desired sharpness is achieved, it is then cut to a set size of 6 cm, where the second piece is then sharpened to the same extent. Once sharpened and cut, a shadow image was taken under the inverse macroscope through a Raspberry Pi system. The pencils were then placed on top of a plate that resides on the Plaster Vibration Machine. Then, in time intervals of two starting from 1, 2, 4, 8, etc, the vibration is set to the lowest frequency and amplitude and run. After the time interval, shadow images were captured of each pencil, which were then uploaded into a MATLAB code that extracts the outline of the image, aligns, and finds the coordinates of the tip to calculate the degradation of each time through 2D collisions. Through each trial, the mass was also taken to measure volume loss with time.

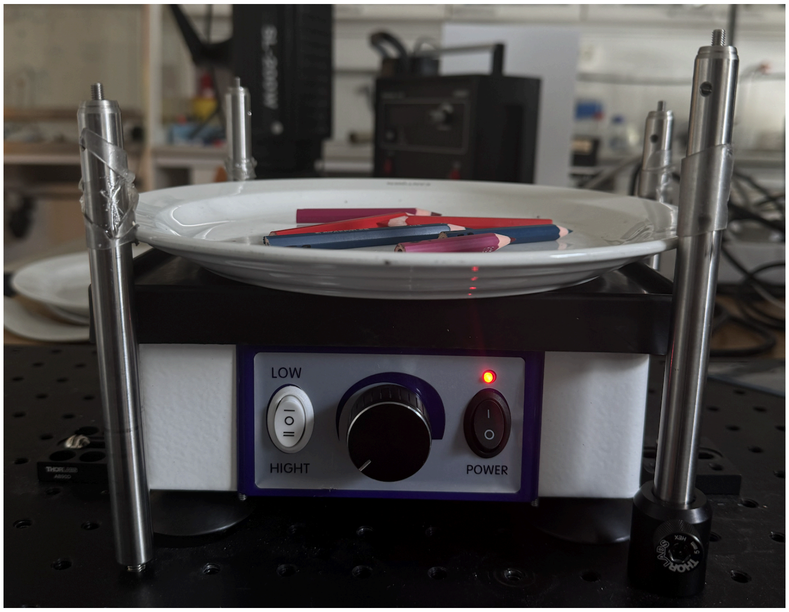


Figure 3: Experimental Setup

3.3 Code

The MATLAB 1 code begins by examining a shadow image of the object (Figure 4), where it is binarized to distinguish the object from the background, to extract its outer boundary. Applying a threshold to enhance the contrast, as well as crop the selection zone of interest for analysis (Figure 5). The cropped image is rotated to align the object vertically for consistent coordinate extraction of pixels of up to 3 to 4 are identified (Figure 6).

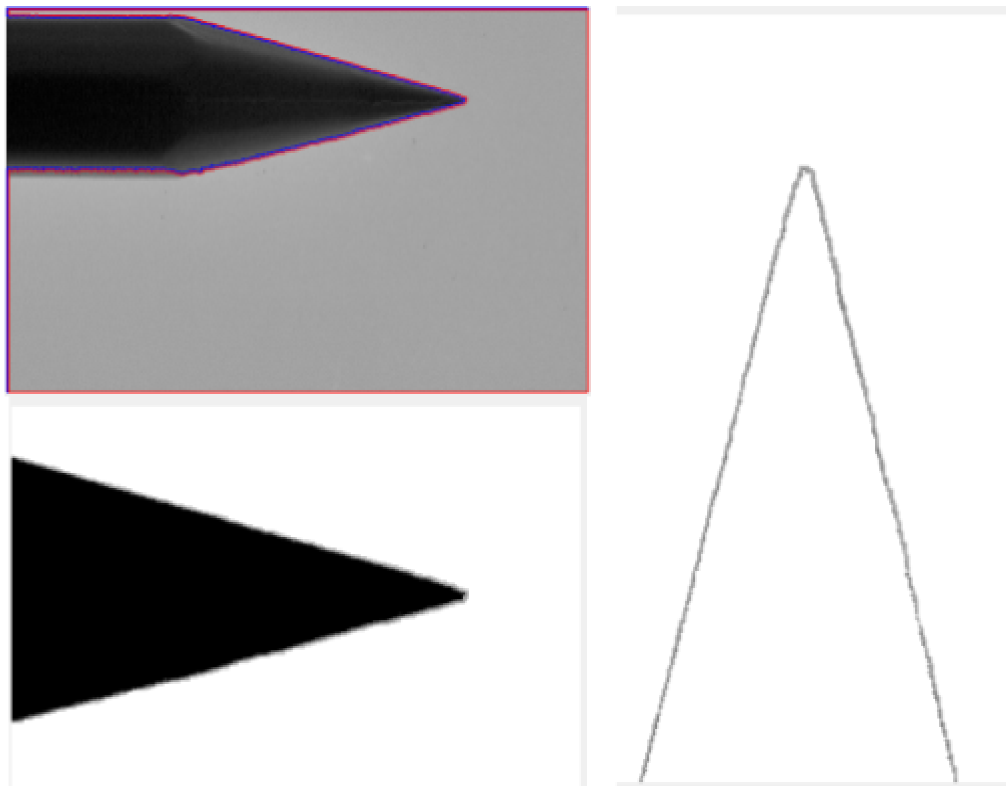


Figure 4-6: Shadow Image, Contrast Image, & Outer Boundary

The coordinates are extracted from the plot, then plotted (Figure 7), which would be divided into left and right sections relative to the object's midpoint. The line of best fit is found separately on each side to represent the object's edges and bisector point (Figure 8).

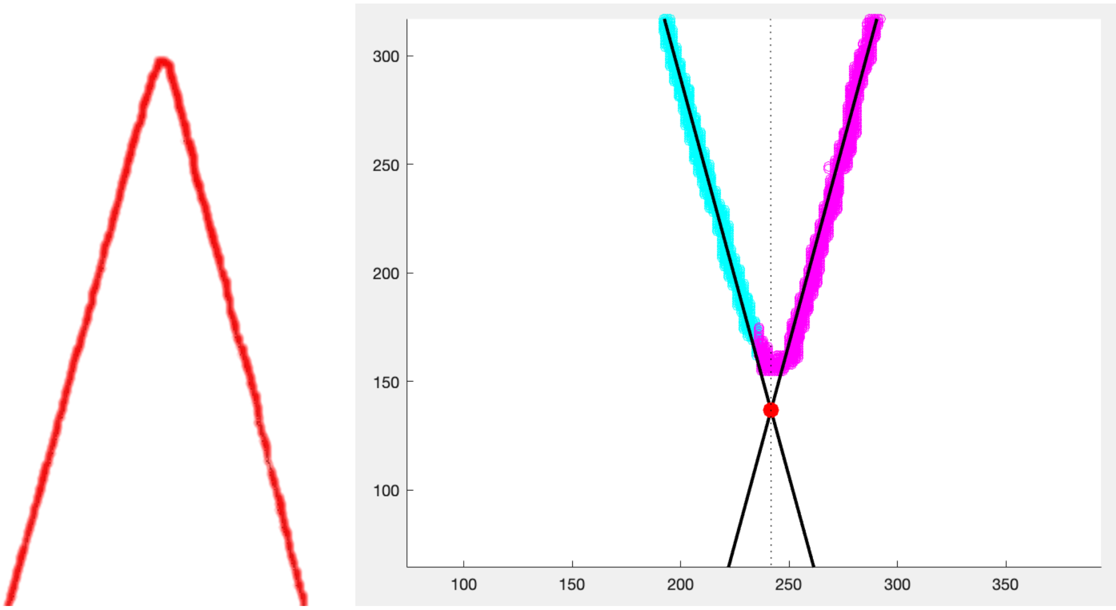


Figure 7 & 8: Graph and Bisector

After obtaining the bisector point, the tip is then graphed and centered for data extraction (Figure 9). It is saved as a CSV file and moved to another MATLAB program.

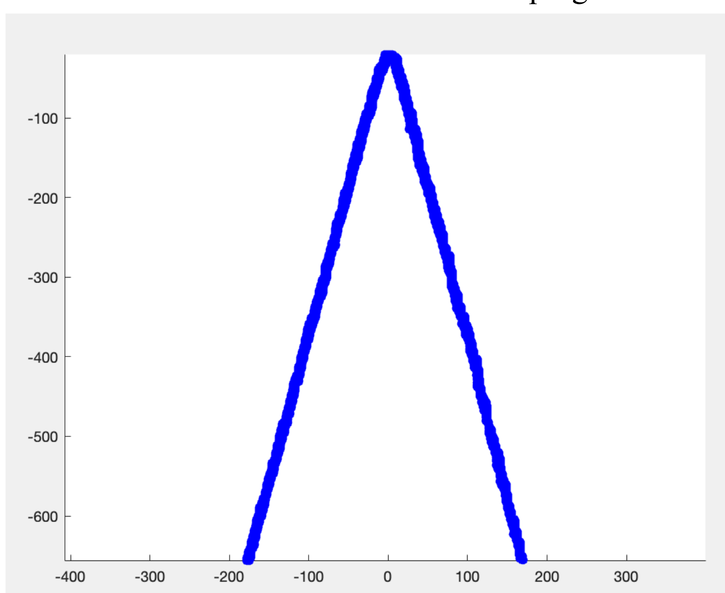


Figure 9: Centered Plot

Taking the CSV file of each trial time interval, it is then added to the MATLAB 2 program called VisAlign to align all extracted data (Figure 10).

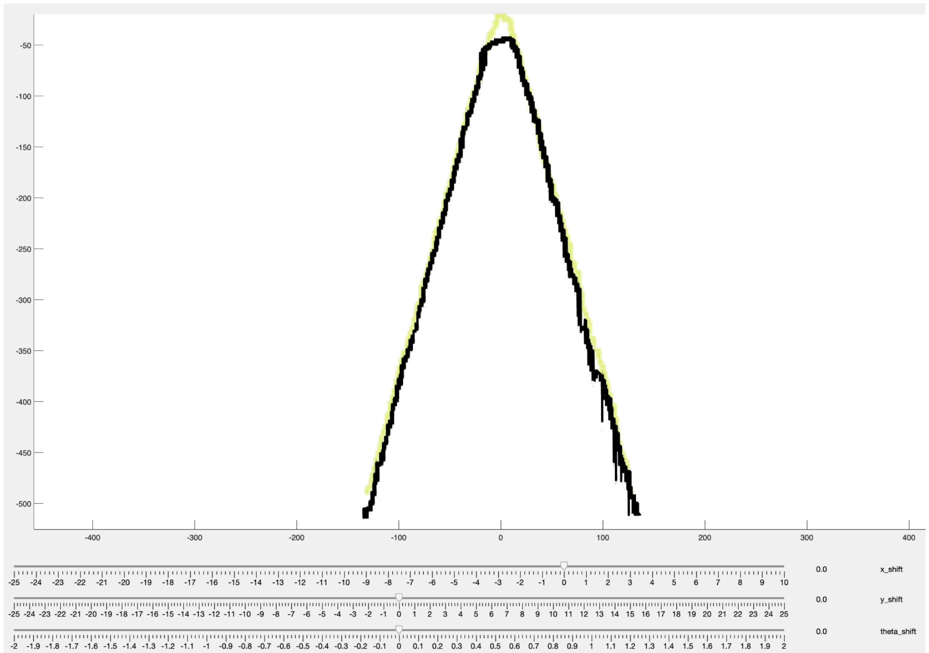


Figure 10: Alignment Plot

Taking all the aligned CSV files of each trial, the change in tip height over time will be aligned and extract maximas of each time interval using MATLAB 3. This is then graphed and measured using Equation 1 on MATLAB 5 to get the log-log graph Δz.

$$\Delta z = z_0 - z \tag{1}$$

MATLAB 4 will take the measured mass at each trial and plot it in a log-log graph while extracting the rate of each mass loss over time using Equation 2.

$$\Delta m = m_0 - m \tag{2}$$

4 | Degeneration of Stingers

This section analyzes how each material degrades after repeated mechanical vibration collisions, focusing on tip sharpness reduction and mass loss over a period of time. These metrics provide insight into whether natural stinger shapes are maintained after extended wear and if optimal geometries are preserved or altered. The material chosen for this experiment includes colored pencil, chalk, and clay, representing different natural stinger structures and mechanical properties.

3.1 Colored Pencils

Colored pencils consist of a pigmented core made from a blend of pigments, such as clay, wax, or oil, encased in a wooden barrel. The core's composition defines its mechanical properties, where wax-based binders produce a softer, smoother core that is easier to sharpen but prone to breakage, while the wooden casing provides structural support but can't fully prevent tip fractures under stress (Minkman 2023). Because of this, colored pencils showed a high susceptibility to tip breakage and jagged edges at the base of the tip during vibration testing and sharpening. This behavior aligns with the core's soft, waxy nature, decreasing durability under repeated mechanical impacts. Uneven stresses and the interaction between the brittle pigmented core and the wooden casing led to asymmetric fracture patterns and sudden tip loss in several trials.

3.2 Chalk

Chalk is a soft, porous rock made mostly of small calcite particles, mixed with clay and organic material (Fabricius 2007). Over time, natural processes like pressure and cementing change its texture and strength, making it brittle. Because of this, chalk tends to crack and break easily when stressed. This matches what we saw in the experiments, where chalk tips often broke or lost mass during vibration tests. Its natural makeup makes it less durable and more prone to wear compared to harder materials. The tip snapped off exactly at the 1-minute mark in each chalk trial tested.

3.3 Clay

Clay was selected to represent a brittle material and became the most unpredictable in the experiment. Using a 3D-printed cone mold, we shaped air-dry clay into pointed structures. However, because the clay was often only partially dry when removed from the mold, uneven drying occurred, especially along the seam where the two halves joined. This seam consistently acted as a weak point, and many samples cracked along this line. As the clay continued to dry, surface cracks formed, increasing its fragility. During vibration testing, tips frequently snapped or crumbled. In some cases, the silhouette of the tip still appeared intact in shadow images, even though material had broken off. This created issues for MATLAB, which uses the outline to measure tip height (ΔZ). Because of this inconsistency, ΔZ data for clay were excluded, though mass loss over time was still recorded.

5 | Results & Discussion

Across all materials tested, patterns of degradation varied significantly depending on material properties. To measure this, the rates of mass loss and tip height reduction (ΔZ) were graphed, which reveal how each material responds to repeated mechanical collisions.

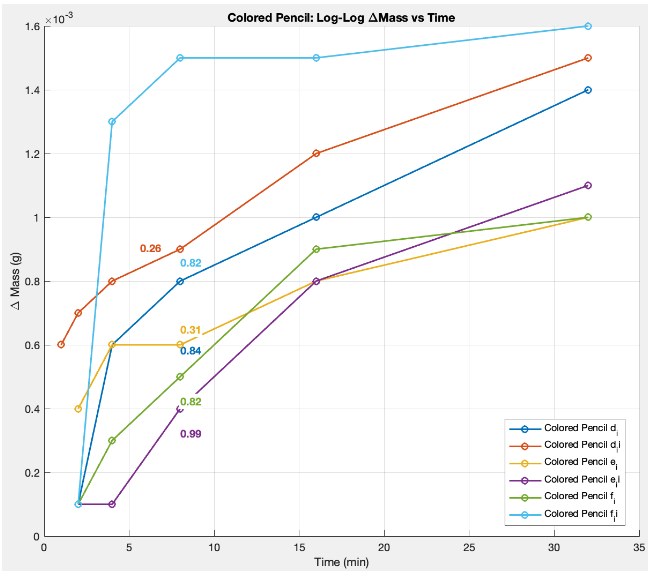
Colored pencils showed the highest average mass loss (0.673) but a moderate ΔZ rate (0.422). This suggests that while the material gradually wears away, the tip retains its shape for longer.

This behavior reflects the nature of its softer material properties, allowing for smoother wear and dulling instead of fracturing. Over time, the pencils tended to round out at the tip rather than snapping.

Chalk, in contrast, had a lower mass loss average (0.438) but the highest ΔZ rate (1.075). This indicates that while not much material was lost overall, the shape of the tip degraded rapidly and mainly through sudden fractures. In several trials, the tips broke cleanly at early time intervals. This behavior aligns with chalk's brittle, porous structure, which doesn't absorb stress well and instead fractures at stress concentrations. It's worth noting that in a few cases, residual chalk from previous trials stuck to the vibrating plate, slightly affecting mass measurements upward. However, the trend of sharp tip loss remained consistent.

Clay presented a different challenge. With an average mass loss of 0.485, it fell between the other two materials. However, due to its brittle cracking and jagged post-fracture edges, the ΔZ measurements were inconsistent and unreliable, and therefore omitted. During trials, the tips frequently snapped along mold seams or micro-cracks formed during drying. Even when the tip appeared intact in the shadow image, material had often broken away, confusing MATLAB's boundary detection. This highlights how unpredictable degradation in brittle materials can be and how they fail differently from those that wear down gradually.

Overall, these results show that not all pointed structures degrade the same way. Chalk tends to fracture sharply, colored pencils dull more slowly, and clay fails abruptly due to internal stress. These behaviors emphasize the role of material composition in determining how a stinger-like structure holds up over time.



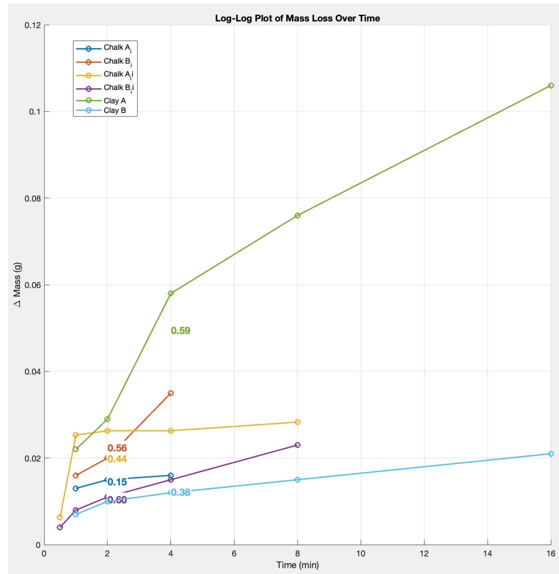
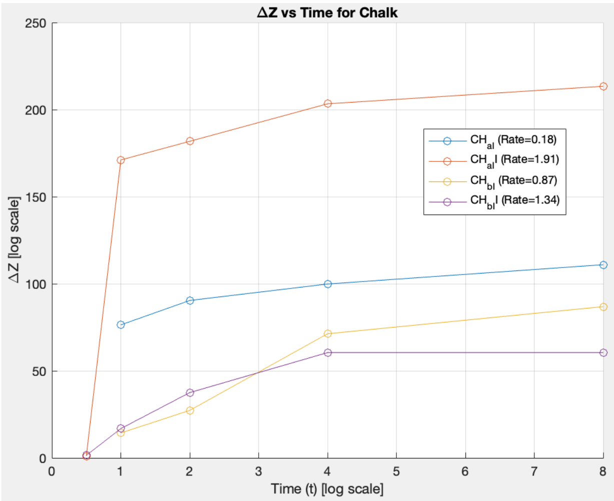


Table 1: Mass Rateddtu

Material	ID	Mass Rate
Colored Pencil	d_i	0.84
Colored Pencil	d_ii	0.26
Colored Pencil	e_i	0.31
Colored Pencil	e_ii	0.99
Colored Pencil	f_i	0.82
Colored Pencil	f_ii	0.82
Chalk	A_i	0.15
Chalk	B_i	0.56
Chalk	A_ii	0.44
Chalk	B_ii	0.60
Clay	A	0.59
Clay	B	0.38



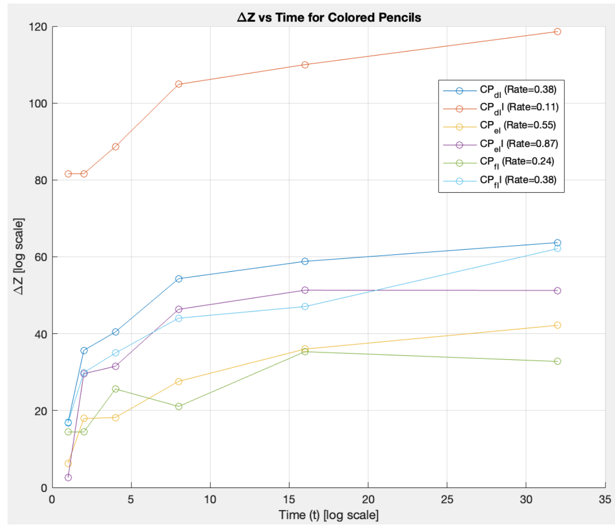


Table 2: ΔZ Rate

Material	ID	ΔZ Rate
Colored Pencil	d_i	0.38
Colored Pencil	d_ii	0.11
Colored Pencil	e_i	0.55
Colored Pencil	e_ii	0.87
Colored Pencil	f_i	0.24
Colored Pencil	f_ii	0.38
Chalk	A_i	0.18
Chalk	B_i	1.91
Chalk	A_ii	0.87
Chalk	B_ii	1.34

Table 3: Average Rate of Materials

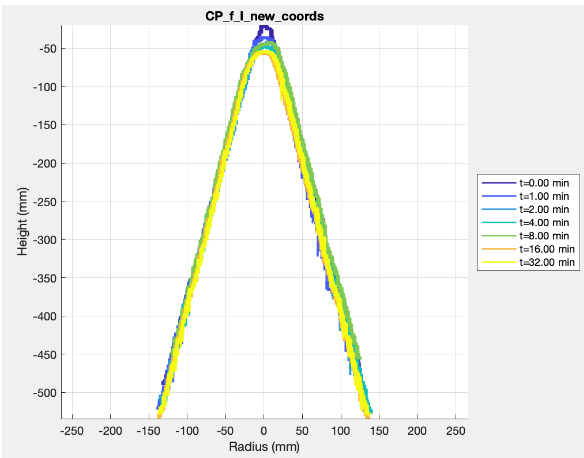
Material	Mass Rate Average	ΔZ Rate Average
Colored Pencil	0.673	0.422
Chalk	0.438	1.075
Clay	0.485	-

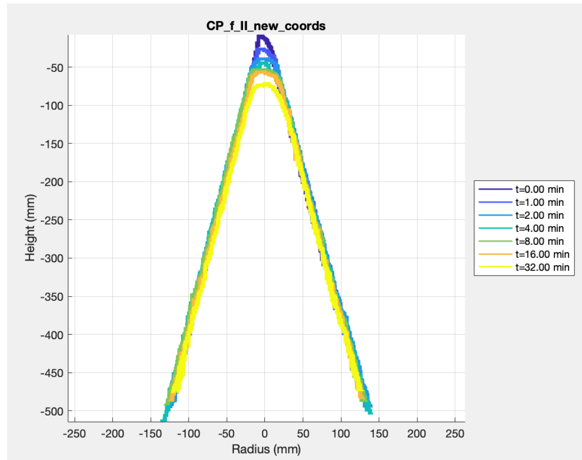
7 | Conclusion

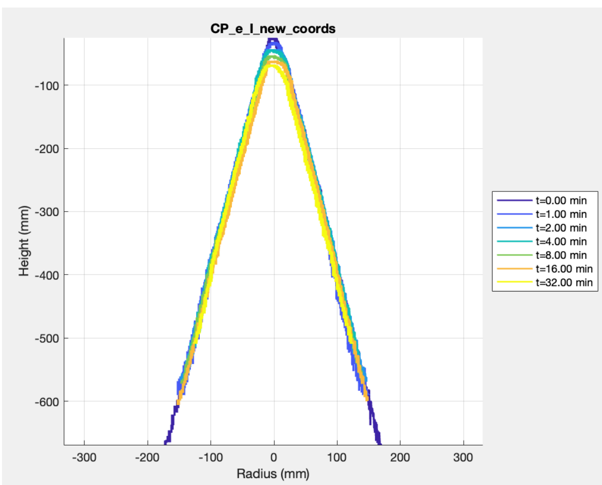
While Quan et al. (2024) showed that natural stingers follow a geometric tapering pattern for maximum penetration, they did not address how these shapes evolve with repeated use. This study extends their work by introducing time and wear as variables, revealing how material properties shape the longevity and performance of pointed structures. Colored pencils, with softer binding material and supportive wood casing, showed steady mass loss and moderate blunting, suggesting predictable and gradual degradation. Chalk, on the other hand, maintained more of its mass but quickly lost tip sharpness due to sudden fractures. Clay's brittle cracking along internal seams further illustrated that degradation is not just about geometry, but also about how material properties handle repeated stress. These results suggest that even if nature defines an "ideal" shape, the real-world effectiveness of a stinger depends heavily on how it holds up under use. Materials that chip, crack, or dull differently can significantly impact long-term function. In bioinspired design, especially in fields like medical needles, drilling tools, or even soft robotics, understanding how a shape wears down is as important as understanding how it performs once. So how does repeated use affect the performance and shape of pointed objects over time? It depends not only on the geometry, but on the wear behavior of the material itself. Whether by fracture, blunting, or unpredictable cracking, degradation follows different paths, and nature always seeks to balance this form with function.

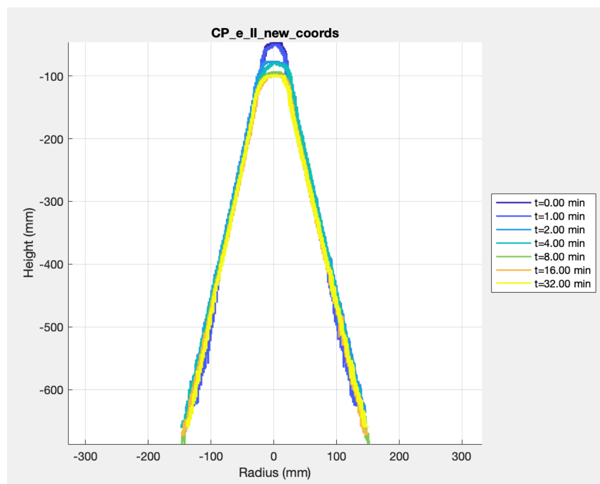
A1 | Graphs

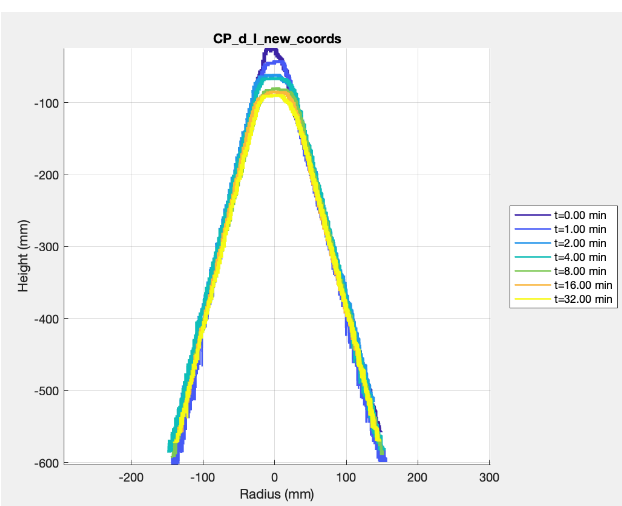
A1.1 Colored Pencil

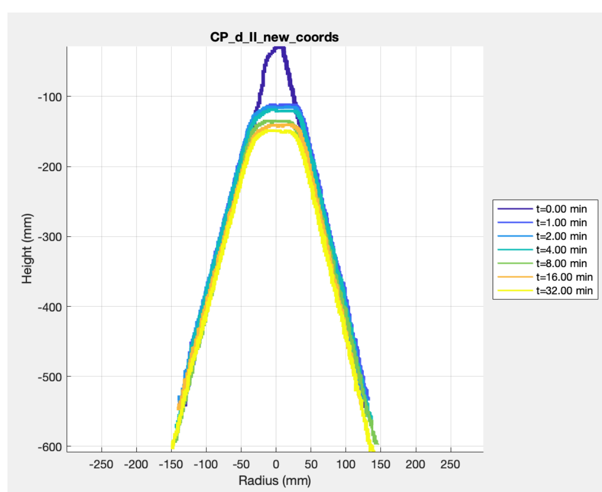




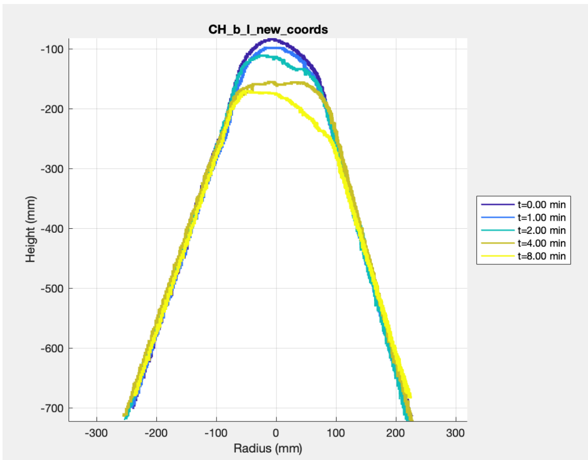


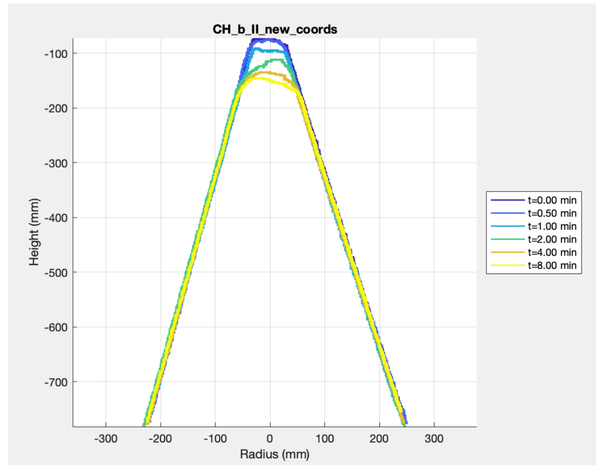


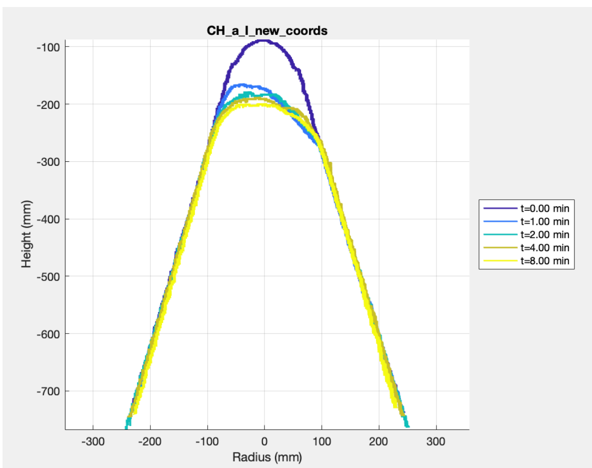


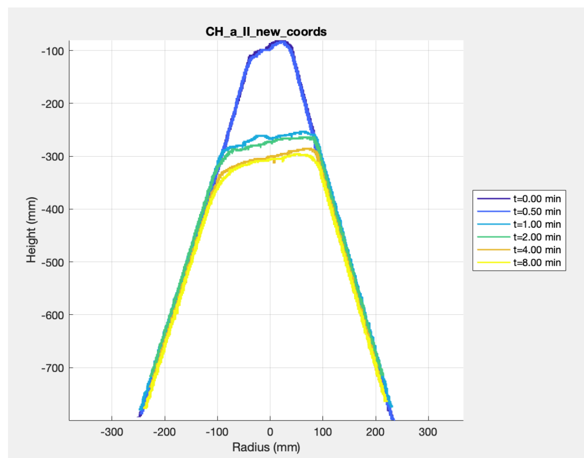


A1.2 Chalk


CH_b_I_new_coords
Height (mm)
Radius (mm)
t=0.00 min
t=1.00 min
t=2.00 min
t=4.00 min
t=8.00 min


CH_b_II_new_coords
Height (mm)
Radius (mm)
t=0.00 min
t=0.50 min
t=1.00 min
t=2.00 min
t=4.00 min
t=8.00 min


CH_a_I_new_coords
Height (mm)
Radius (mm)
t=0.00 min
t=1.00 min
t=2.00 min
t=4.00 min
t=8.00 min


CH_a_II_new_coords
Height (mm)
Radius (mm)
t=0.00 min
t=0.50 min
t=1.00 min
t=2.00 min
t=4.00 min
t=8.00 min

A2 | MATLAB Code

A2.1 MATLAB 1

```
clear all
close all

folder_name = '/Users/shristhipant/Documents/MATLAB/2D_Gladiator/Colored_Pencils';
f_name = 'CPf_I_t0.png'; % update filename
full_name = fullfile(folder_name, f_name);

img = imread(full_name);
%img = im2gray(img);
img = img(:,:,1);

imshow(img);
imhist(img)
img_bin = imbinarize((img));
boundaries = bwboundaries(img_bin,8);
bounds_sizes = cellfun(@numel, boundaries);
length(bounds_sizes);
[~, maxIndex] = max(bounds_sizes);
profile_full = boundaries{maxIndex};

u = profile_full(:,2);
v = profile_full(:,1);
   boundaryMask = poly2mask(u, v, size(img,1), size(img,2) ); % Convert boundary to mask
   dilatedMask = imerode(boundaryMask, strel('disk', 5, 8));
   dilatedBoundary = bwboundaries(dilatedMask);
   dilated_u = dilatedBoundary{1}(:, 2); % x coordinates (u)
   dilated_v = dilatedBoundary{1}(:, 1); % y coordinates (v)
imshow(img)
hold on
plot(u, v, 'b-', 'LineWidth', 2);
plot(dilated_u, dilated_v, 'r-', 'LineWidth', 2);


%%
img_clean = img;
img_clean(dilatedMask) = 255;
imshow(img_clean)
img1 = lim_img(img_clean, 110,200); % 110,200

% Crop image- click top left and bottom right points
close all
imshow(img1)
[limx, limy] = ginput(2);
img2 = imcrop(img1,[limx(1) limy(1) diff(limx) diff(limy)]);
img2(:,end-5:end) = 255;
imshow(img2)

%%
img3 = img2;
img3(img2==0)=255;
img3 = imrotate(img3,90);
imshow(img3)
[aa, bb] = find(img3~=255);

%% Plot Extracted Coordinates
close all
figure()

imshow(img3);
hold on
scatter(bb,aa,25,'o','filled','MarkerFaceColor','r','markerfacealpha',0.05);
```

```
%%
close all
base_snip = 1.8;
cone_base = max(aa)/base_snip;
mid_point_appr = min(bb)+(max(bb)-min(bb))/2;

aaL = aa(bb<mid_point_appr);
bbL = bb(bb<mid_point_appr);
aaR = aa(bb>mid_point_appr);
bbR = bb(bb>mid_point_appr);

lin_l_x = bbL(aaL<cone_base);
lin_l_y = aaL(aaL<cone_base);
lin_r_x = bbR(aaR<cone_base);
lin_r_y = aaR(aaR<cone_base);


scatter(bb,aa)
hold on
axis equal
scatter(bbL,aaL,'b')
scatter(lin_l_x,lin_l_y,'c')
scatter(lin_r_x,lin_r_y,'m')

pl = fitLinePCA(lin_l_x,lin_l_y);
m1 = pl(1);
b1 = pl(2);
xl_fit = linspace(min(lin_l_x), numel(aa)/2, 1000);
yl_fit = m1 * xl_fit + b1;
plot(xl_fit, yl_fit, 'k-', 'LineWidth', 2);
pr = fitLinePCA(lin_r_x,lin_r_y);
m2 = pr(1);
b2 = pr(2);
xr_fit = linspace(min(lin_l_x), numel(aa)/2, 1000);
yr_fit = m2 * xr_fit + b2;
plot(xr_fit, yr_fit, 'k-', 'LineWidth', 2);
%
xc = - (b1 - b2)/(m1- m2)
yc = m1 * xc + b1 % pl(1)*pr(2) - pr(1)*pl(2) / (pl(1) - pr(1));
scatter(xc,yc,100,'r','filled')
%
theta = 0.5*( atan(m1) + atan(m2))
m_mid = tan(pi/2 + theta); % (m1 - m2) / (sqrt( (m1-m2)^2 + (1+m1*m2) ) + (1+m1*m2) );
x_mid = linspace(min(lin_l_x), numel(aa)/2, 1000);
y_mid = m_mid* (x_mid - xc) + yc;
plot(x_mid,y_mid,'k:')
ylim([min(lin_l_y),yc+100])


%% Angle correction based on bisector slope
coordX = bb - xc;
coordY = -(aa - yc);
[rX, rY] = rotateCoords(coordX, coordY, theta);
[rY, sort_idx] = sort(rY);
rX = rX(sort_idx);
close all
scatter(rX, rY,'b', 'filled');
axis equal

%% Save coords
folder_name = './a'; % Update folder name
subfolder = 'coords_';
```

```matlab

folder_path = fullfile(folder_name,subfolder) ;
if ~exist(folder_path, 'dir')
   mkdir(folder_path);
end
writematrix([rX', rY'], strcat(folder_path,f_name,'.csv'));



%%%%%%%%%%%%%%%%%%%%%%%%% FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%%%%
%% threshold/ contrast
function img_thresh = lim_img(image_raw, lower_lim,upper_lim)
   img_thresh = image_raw;
   img_thresh(image_raw<lower_lim)=0;
   img_thresh(image_raw>upper_lim) = double(intmax(class(image_raw)));
end
%%
function p = fitLinePCA(xx, yy)
   if length(xx) ~= length(yy)
       error('check size(xx) and size(yy)');
   end

   data = [xx(:), yy(:)];
   data_centered = data - mean(data);
   [coeff, ~, ~] = pca(data_centered);
   direction = coeff(:,1);
   m = direction(2) / direction(1);  % Slope of the line
   mean_x = mean(xx);
   mean_y = mean(yy);
   b = mean_y - m * mean_x;          % Intercept of the line
   p = [m, b];
end

function [xrot, yrot] = rotateCoords(x, y, theta)
   R = [cos(theta), -sin(theta); sin(theta), cos(theta)];
   rotated_coords = R * [x(:)'; y(:)'];
   xrot = rotated_coords(1, :);
   yrot = rotated_coords(2, :);
end

```

A2.1 MATLAB 3

```
clear; close all; clc;

% Root folder path
root_folder = './Coords';
if ~isfolder(root_folder)
    error('Folder "%s" not found.', root_folder);
end

% List subfolders
part_folders = dir(root_folder);
part_folders = part_folders([part_folders.isdir] & ~startsWith({part_folders.name}, '.'));

% Preallocate
dz_all = {};
time_all = {};
part_labels = {};
material_types = {};
maxima_all = {}; % NEW: stores maxima per trial

% Specify folders where time is in seconds and must be converted to minutes
seconds_to_minutes = {'CH_a_II', 'CH_b_II'};

for f = 1:length(part_folders)
    part_name = part_folders(f).name;
    fprintf('Processing folder: "%s"\n', part_name);

    full_path = fullfile(root_folder, part_name);
    csv_files = dir(fullfile(full_path, '*.csv'));
    if isempty(csv_files)
        fprintf('  No CSV files found, skipping.\n');
        continue;
    end

    [~, idx] = sort({csv_files.name});
    csv_files = csv_files(idx);

    profiles = {};
    times = zeros(1, length(csv_files));
    handles = gobjects(1, length(csv_files));

    convert_to_min = any(contains(part_name, seconds_to_minutes));
    fprintf('  Convert to minutes: %d\n', convert_to_min);

    for i = 1:length(csv_files)
        file_path = fullfile(full_path, csv_files(i).name);
        data = readmatrix(file_path);
        if size(data, 2) < 2
            warning('  File %s skipped: less than 2 columns', csv_files(i).name);
            continue;
        end

        rX = data(:,1);
        rY = data(:,2);
        profiles{i} = [rX, rY];

        t = regexp(csv_files(i).name, '\d+(\.\d+)?', 'match');
        if isempty(t)
            times(i) = i;
        else
            times(i) = str2double(t{1});
        end
```

```
        if convert_to_min
            times(i) = times(i) / 60;
        end
    end

    valid = ~cellfun(@isempty, profiles);
    profiles = profiles(valid);
    times = times(valid);

    [times, sortIdx] = sort(times);
    profiles = profiles(sortIdx);

    % Plot each profile in time order (single trial graphs)
    figure('Name', part_name); hold on;
    cmap = parula(length(profiles));
    handles = gobjects(1,length(profiles));
    for i = 1:length(profiles)
        rX = profiles{i}(:,1);
        rY = profiles{i}(:,2);
        handles(i) = plot(rX, rY, 'Color', cmap(i,:), 'LineWidth', 1.2);
    end
    title(strrep(part_name, '_', '\_'));
    xlabel('Radius (mm)');
    ylabel('Height (mm)');
    legend_labels = arrayfun(@(t) sprintf('t=%.2f min', t), times, 'UniformOutput', false);
    legend(handles, legend_labels, 'Location', 'eastoutside');
    axis equal; grid on; hold off;

    % Calculate ΔZ relative to first profile
    ref_X = profiles{1}(:,1);
    ref_Y = profiles{1}(:,2);
    dz_trial = zeros(1, length(profiles));
    maxima_trial = zeros(1, length(profiles)); % NEW: Max height for each trial

    for i = 1:length(profiles)
        curr = profiles{i};
        interp_Y = interp1(curr(:,1), curr(:,2), ref_X, 'linear', 'extrap');
        dz = ref_Y - interp_Y;
        dz_trial(i) = mean(abs(dz), 'omitnan');

        % Get max height of current profile
        maxima_trial(i) = max(curr(:,2), [], 'omitnan');
    end

    dz_all{end+1} = dz_trial;
    time_all{end+1} = times;
    part_labels{end+1} = part_name;
    maxima_all{end+1} = maxima_trial; % Store maxima

    if contains(part_name, 'CH')
        material_types{end+1} = 'CH';
    elseif contains(part_name, 'CP')
        material_types{end+1} = 'CP';
    else
        material_types{end+1} = 'Unknown';
    end

    % Print maxima
    fprintf('  Max heights for %s:\n', part_name);
    for i = 1:length(times)
        fprintf('    Time %.2f min: Max Height = %.2f mm\n', times(i), maxima_trial(i));
    end
end

% Function to clean label (remove "_new_coords")
cleanLabel = @(str) erase(str, '_new_coords');
```

A2.1 MATLAB 4

```
% Colored Pencil Data
data_cp = {
    'Colored Pencil','d_i',[0 1 2 4 8 16 32],[2.4424 2.4424 2.4423 2.4418 2.4416 2.4414 2.441];
    'Colored Pencil','d_ii',[0 1 2 4 8 16 32],[1.965 1.9644 1.9643 1.9642 1.9641 1.9638 1.9635];
    'Colored Pencil','e_i',[0 1 2 4 8 16 32],[2.7392 2.7392 2.7388 2.7386 2.7386 2.7384 2.7382];
    'Colored Pencil','e_ii',[0 1 2 4 8 16 32],[3.286 3.286 3.2859 3.2859 3.2856 3.2852 3.2849];
    'Colored Pencil','f_i',[0 1 2 4 8 16 32],[2.7756 2.7756 2.7755 2.7753 2.7751 2.7747 2.7746];
    'Colored Pencil','f_ii',[0 1 2 4 8 16 32],[3.2986 3.2987 3.2985 3.2973 3.2971 3.2971 3.297];
};

% Log-Log Plot
figure;
hold on;
colors = lines(length(data_cp));
title('Colored Pencil: Log-Log \DeltaMass vs Time');

for i = 1:size(data_cp,1)
    label = [data_cp{i,1} ' ' data_cp{i,2}];
    time = data_cp{i,3};
    mass = data_cp{i,4};
    delta_mass = mass(1) - mass;

    mask = delta_mass > 0 & time > 0;
    loglog(time(mask), delta_mass(mask), 'o-', ...
        'DisplayName', label, ...
        'Color', colors(i,:), ...
        'LineWidth', 1.5);

    % Fit and annotate slope
    log_t = log10(time(mask));
    log_dm = log10(delta_mass(mask));
    coeffs = polyfit(log_t, log_dm, 1);
    slope = coeffs(1);

    mid_x = 10^(mean(log_t));
    mid_y = 10^(mean(log_dm));
    text(mid_x, mid_y, sprintf('%.2f', slope), ...
        'Color', colors(i,:), ...
        'FontSize', 10, ...
        'FontWeight', 'bold', ...
        'BackgroundColor', 'white', ...
        'Margin', 1);
end
xlabel('Time (min)');
ylabel('\Delta Mass (g)');
legend('Location', 'best');
grid on;

% Linear Plot (Recommended for visibility)
figure;
hold on;
title('Colored Pencil: Linear \DeltaMass vs Time');
for i = 1:size(data_cp,1)
    label = [data_cp{i,1} ' ' data_cp{i,2}];
    time = data_cp{i,3};
    mass = data_cp{i,4};
    delta_mass = mass(1) - mass;
    plot(time, delta_mass, 'o-', 'DisplayName', label, ...
        'Color', colors(i,:), 'LineWidth', 1.5);
end
xlabel('Time (min)');
ylabel('\Delta Mass (g)');
legend('Location', 'best');
grid on;
```

A2.1 MATLAB 5

```matlab
% CH Data
ID = { 'CH_a_II','CH_a_II','CH_a_II','CH_a_II','CH_a_II','CH_a_II',...
       'CH_a_I','CH_a_I','CH_a_I','CH_a_I','CH_a_I',...
       'CH_b_II','CH_b_II','CH_b_II','CH_b_II','CH_b_II','CH_b_II',...
       'CH_b_I','CH_b_I','CH_b_I','CH_b_I','CH_b_I'}';

t = [ 0,0.5,1,2,4,8,...
      0,1,2,4,8,...
      0,0.5,1,2,4,8,...
      0,1,2,4,8 ]';

DeltaZ = [ 0,1.08,171.08,181.79,203.32,213.39,...
           0,76.38,90.4,99.85,110.94,...
           0,1.47,16.82,37.5,60.47,60.47,...
           0,14.32,27.29,71.29,86.86 ]';

% Create table
data = table(ID,t,DeltaZ);

% Plotting
figure; hold on;
IDs = unique(data.ID);
colors = lines(length(IDs));
legendEntries = {};

for i = 1:length(IDs)
    idx = strcmp(data.ID, IDs{i});
    x = data.t(idx);
    y = data.DeltaZ(idx);

    % Only keep values > 0 for log-log
    valid = (x > 0) & (y > 0);
    x_log = x(valid);
    y_log = y(valid);

    if length(x_log) < 2
        continue
    end

    % Plot log-log
    loglog(x_log, y_log, '-o', 'Color', colors(i,:));

    % Compute log-log rate
    rate = (log(y_log(end)) - log(y_log(1))) / (log(x_log(end)) - log(x_log(1)));

    % Add to legend
    legendEntries{end+1} = sprintf('%s (Rate=%.2f)', IDs{i}, rate);
end

xlabel('Time (t) [log scale]');
ylabel('\DeltaZ [log scale]');
title('Log-Log Plot of \DeltaZ vs Time for Chalk with Rates');
legend(legendEntries, 'Location','eastoutside');
grid on;
hold off;
```

Bibliography

Fabricius, I. L. (2007). Chalk: Composition, diagenesis and physical properties. *Bulletin of the Geological Society of Denmark*, *55*, 97–128. https://doi.org/10.37570/bgsd-2007-55-08

Minkman, L. (2023, April 14). *How are colored pencils made and the key ingredients involved*. Drawing tips and tricks. https://luukminkman.com/en/blog/how-are-colored-pencils-made/

Quan, H., Liang, X., Zhang, X., Meyers, M. A., McMeeking, R. M., & Arzt, E. (2024). The shape of nature's Stingers revealed. *Proceedings of the National Academy of Sciences*, *121*(7). https://doi.org/10.1073/pnas.2316320121