

REU-HYPER 2019

IMAGE PROCESSING AND DATA ANALYSIS OF FLAME PROPAGATION IN A COMBUSTION CHAMBER

Camille Phillips¹, Suhyeon Park²

¹Georgia Southern University, Statesboro, GA

²University of Central Florida, Orlando, FL

ABSTRACT

The purpose of this study was to use high-speed flame propagation images to calculate the flame speed of the given natural gas mixtures at atmospheric pressure within a constant volume chamber. Using flame propagation images from three combustion experiments conducted at different initial pressures, flame speeds were found, analyzed and compared. The change in radii of the flames in each of the high-speed flame image sets over time were found. Combustion simulations were run at the designated initial pressures to find the density of burnt to unburnt gas for each of the trials. These density ratios, along with the radius growth measurements, were incorporated to find the flame speeds of the mixture at the desired initial pressures. The images were processed using MATLAB and flame speed calculations were made under an assumption of constant pressure.

Significant background research was conducted on various coding methods in relation to flame propagation, image processing, and fluctuating area measurement techniques. However, the coding examples found were application-specific and non-versatile, therefore not applicable to the current study. The finalized code employs a circular masking area that eliminates “noise” in the outlying area. The code has been integrated to fit multiple data sets with different initial pressures and therefore slightly varying density ratios. A combustion simulation program called GasEQ was employed to find the density ratio of burnt to unburnt gas which is essential to finding flame speed. The flame speeds of the data sets were calculated and compared in a plot. Work continues to be done on analyzing the flame speed data. Currently, the uncertainty of the flame speed results is being studied. Further efforts include calculating stretch rate of the flames. In the future, the code could be modified to run through other programs such as Python for more widespread usage.

v_u velocity of unburned gas
 ρ_b density of burnt gas
 ρ_u density of unburnt gas

1. INTRODUCTION

Combustion is a fundamental aspect of power generation and propulsion systems such as gas turbines and rocket engines. Insight on flame speed allows for the understanding of many combustion characteristics. Flame speed is a measurement that describes how a flame propagates into a section of unburned mixture at certain temperatures and pressures. It is called laminar burning velocity (LBV) when the flame is in laminar regime [1]. Flame speed can be measured by expansion rate of a flame during a combustion reaction. Flame speed measurements will be discussed in this report.

The faster a flame grows, the larger the area a flame can achieve, leading to an increase in combustion activity. The ability to calculate flame speed of a certain fuel allows for the subsequent calculation of efficiency and can provide time expectations on how long the fuel will take to burn into the unburnt mixture. Figure 1 provides a sample of the high-speed flame images and illustrates the direction of growth. Equations (1) and (2) describe the varying flame speed and the flame radius growth rate [2]. The flame speed is given by the product of the flame radius time derivative and the density ratios of burnt and unburnt gas.

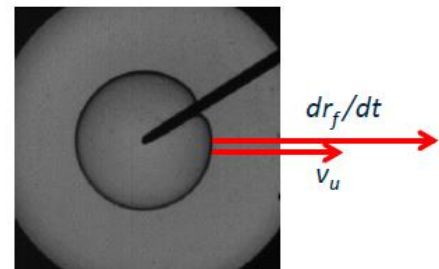


FIGURE 1: FLAME GROWTH SAMPLE

NOMENCLATURE

A_f flame area
 r_f flame radius
 S_u flame speed

$$\frac{dr_f}{dt} = v_u + S_u \quad (1)$$

$$S_u = \frac{\rho_b}{\rho_u} \frac{dr_f}{dt} \quad (2)$$

While flame speed is relatively easily measured using time-varying pressure data when a flame grows in a regular, spherical fashion, problems arise when the growth becomes irregular or “wrinkled”. In such situations, the radius estimated from pressure data can no longer be used to calculate the flame speed. The purpose of this study was to employ image processing tools in MATLAB to combat this issue. Similarly, this study focused on obtaining the effective radii associated with the resulting areas found so as to discern the flame speeds of the given mixture at different initial pressures as shown in equation (3). Once obtained, the flame speeds were used to model and analyze the combustion characteristics of each pressure trial.

$$r_{f,eff} = \sqrt{\frac{A_f}{\pi}} \quad (3)$$

The flame growth experiments which produced the high-speed flame propagation images found in this study were performed in a constant volume combustion chamber. However, the instantaneous pressure can be assumed as spatially uniform near the flame surface. Although the natural gas mixture remained constant for all three trials, the initial pressure of the mixture was changed for all three trials. Through the use of GasEQ, many qualities before and after the combustion were calculated and observed, to include the densities of burnt and unburnt gases. While the density ratio of unburnt gas to burnt gas varied slightly per trial, it was generally around the 0.11-0.12 range.

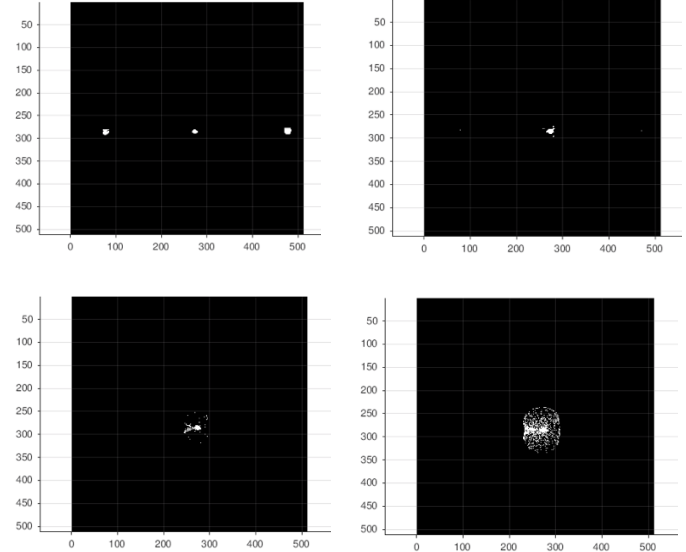
2. MATERIALS AND METHODS

MATLAB Code

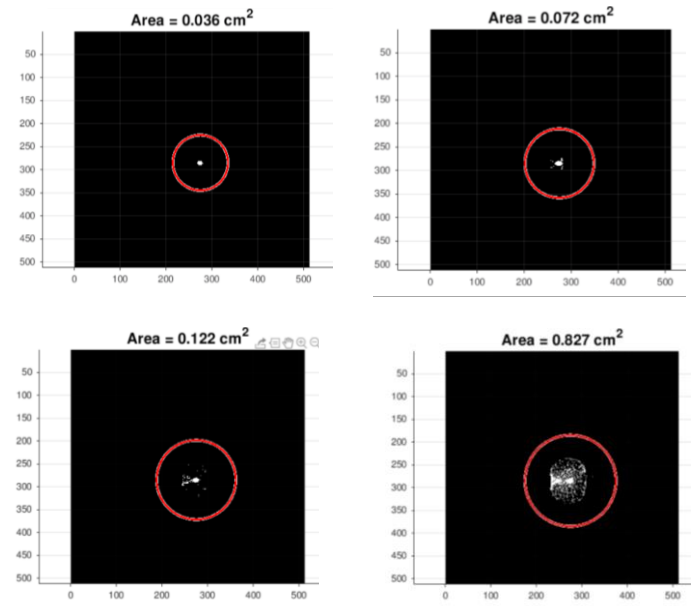
The raw images of the flame growth are shown in Figures 2-5. The images show unfiltered flame emission of flames which were recorded using a Phantom V12 high-speed camera at a rate of 3000 frames per second with 512×512 resolution. The outside noise is most prevalent in the initial spark image, with two sections of white pixels on opposite edges of the frame. The goal of this study was to eliminate the outside areas as seen in Figure 2 to accurately calculate the main flame area.

When approaching the fluctuating area calculation, the first step taken was to remove noise by ignition spark and reflections around the main area of the flame. To combat this, a masking feature was developed that would follow the growth of the center flame. The mask was made to be an expanding circle and allowed only the areas within the growing radius of the circle to be calculated, excluding the outlying noise. A time-varying value was employed to increase the radius as time passed and was consistently modified to best fit the area growth. Once the mask was created, the images were converted to black-and-white binary images so that the white pixels (the center flame area) could be counted and converted to an area measurement in cm^2 .

The masking feature is visualized in (Figures 6-9): the white colored pixels are the growing area and the red circle is the masking function matching the area growth. In the completed coded section, the area and mask figure windows were suppressed, and the only outputs are the radii lengths.



FIGURES 2-5: FLAME GROWTH VISUALIZATION: RAW IMAGES (FRAMES 1, 3, 5, 7)



FIGURES 6-9: VISUALIZATION OF MASK RADIUS GROWTH TO MATCH FLAME AREA GROWTH (FRAMES 1, 3, 5, 7)

By differentiating the radius length over time data, the radius growth over time was found. Subsequently, finding the product of radius growth and the density ratio of burnt and unburnt gas specific to the experiment trials produced the flame speed.

It should be noted that only the first thirty high speed flame propagation images (out of one hundred) from each data set were analyzed. This is because the main goal of this study was to understand the initial flame growth immediately after ignition. After the first 30 frames, the flame begins to exceed the optical access boundary and become difficult to draw any conclusions from.

Flame Speed Trials

GasEQ was used to simulate the natural gas combustion. The mixture contained hydrocarbon mixture as fuel along with oxidizer and diluent gas. The initial pressures of Tests 01-03 were 2.079, 1.000 and 0.659 atm. The initial temperature of each trial was 50 °C. When the simulation was run, the density ratios for each trial were found and recorded in Table 1. Problem type was set to adiabatic constant pressure. The amount of pressure rise for the selected region of data (initial 30 frames or <10 ms) is very small compared to the final pressure after combustion, and the pressure can be assumed as constant.

Calibration of the flame images produced a conversion factor of 0.020 cm per pixel, which remained constant throughout the trials. Using the density ratios and by running the high-speed flame images from the respective trials through the code, the radius growth and subsequent flame speed measurements were recorded.

3. RESULTS AND DISCUSSION

In order to obtain the varying flame speed (S_u) of the given mixture, the overall change in radius over time ($\frac{dr_f}{dt}$) must be found and scaled with the density ratio of burnt and unburnt gas ($\frac{\rho_b}{\rho_u}$). The GasEQ simulations produced Table 1, which displays the density ratios for the three data sets analyzed.

TABLE 1: GASEQ TRIALS COMBUSTION MIXTURE DENSITY OF BUNRT TO UNBURNT GAS RATIOS

	Density (kg/m ³)			Pressure (atm)
	Reactants	Products	Ratio	
Test 01	2.5784	0.3063	0.11882	2.079
Test 02	1.2396	0.1493	0.12051	1.000
Test 03	0.8181	0.0994	0.12152	0.659

Once the respective density ratios were found, they were incorporated into the MATLAB code. The product of the density ratio and radius growth over time for each trial produced Figure 10. Several regions of Figure 10 are of importance. To start with, after the spark is generated and the flame is initiated, for Tests 01 and 02, the flame speed drops suddenly before climbing up to the first peak. This shows the decreasing volume of spark by heat dissipation immediately after the spark. As flame start to propagate, the flame speed rises back to positive value. In Test 03, at 0.659 atm, the flame speed does not decrease at flame initiation.

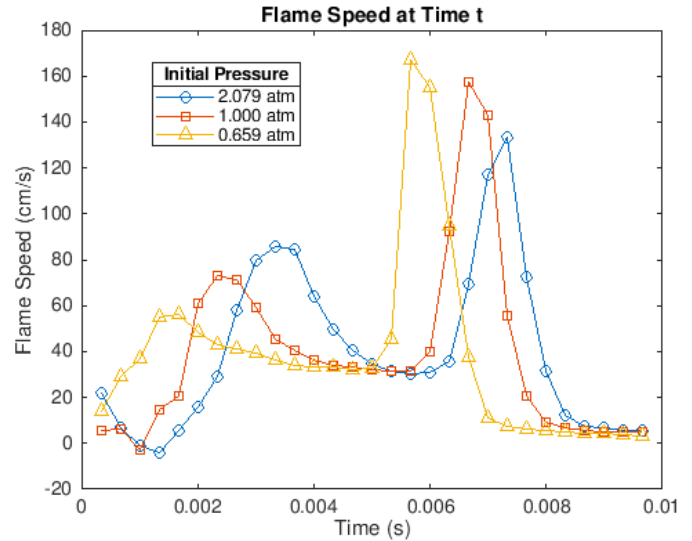


FIGURE 10: FLAME SPEED DATA FRAMES [1:30] AT DIFFERENT INITIAL PRESSURES

Analyzing the first peaks of each trial as a group, the highest-pressure trial shows the greatest flame speed in the first peak in its curve, and the lowest pressure trial shows the lowest flame speed at its first peak. After the first initial peak in flame speed in all three trials, all of the curves level out to about the same speed (40 cm/s), before climbing to the second spike in flame speed. Again, looking at the second spike section of the trials together, Test 03, at the lowest initial pressure, shows the highest flame speed at its second peak, followed by Test 02 and Test 01, respectively. Even though the causes of the peaks are not clearly identified yet, the second peak is more consistent with general expectation by looking at the relation of second peak to initial pressure. Typically, flame speed decreases with increasing pressure when temperature is held constant [6]. A more compressed gas mixture results in lower spatial propagation which is more than compensation of the faster chemical reaction [6]. Finally, the curves all level out to the 10 cm/s range when the flame reaches the surface of the chamber and the flame is quenched. Further analysis of the results is required to understand the comparative flame speed behavior.

4. CONCLUSION

Flame propagation measurement helps to better understand combustion because it is a major factor in determining the reactivity, diffusivity and heat transfer of a combustion mixture. Calculating the flame speed of fuels allows for the modeling of the flames themselves, which leads to higher efficiency in engine design and predictive modeling of the combustion. The ability to manipulate combustion reactions using an understanding of flame speed could potentially lead to advances in fuel performance and capabilities for the future.

In this study, there were two subtasks for the overall goal, flame speed measurement; 1) developing flame tracking algorithm with image processing, and 2) combustion simulation to learn more about the combustion process. The first task to

tackle was the flame tracking issue. A series of high-speed flame propagation images were processed through MATLAB to track the flame and calculate the flame radius growth over time. To achieve this, a masking feature was created that formed an expanding radius to encompass only the desired central flame growth. The developed MATLAB code successfully analyzed the images with low signal intensity and calculated radius growth rate of flames while rejecting outlying noise. The next task was combustion simulation. Using a combustion simulation tool GasEQ, combustion of the fuel mixture at three different initial pressures was simulated, and the density ratio of burnt to unburnt gas was found. This ratio was applied to the MATLAB code and allowed for flame speed measurements to be found of the three separate tests.

Further efforts surrounding this method of flame speed analysis include calculating measurement uncertainty. Attempts have been made at such a calculation, but a deeper understanding of how to estimate the uncertainty with respect to a differentiated function is required. Similarly, calculating stretch rate of the flames analyzed could provide more information on the flame speed behavior demonstrated in this study [4, 5]. In the future, this MATLAB coding method could be integrated to evaluate flame image sets of other combustible mixtures. To accomplish this, modifications would have to be made to the masking feature to best match the radius growth to that of the specific fuel flame growth so as to continue to eliminate the outlying noise. If this method is utilized in the future, a more reliable combustion simulation tool such as Cantera could enhance the result. The MATLAB code could also be altered to run through other programs such as Python or C++ for more widespread usage.

ACKNOWLEDGEMENTS

Dr. Subith Vasu, Gihun Kim, Robert Burke

REFERENCES

- [1] Vasu, Subith. "Laminar burning velocities of high-performance fuels relevant to the Co-Optima Initiative." PhD 19PFL-0129. University of Central Florida, Orlando, FL. 2019.
- [2] M.P. Burke, X. Qin, Y. Ju, F.L. Dryer, 2007, "Measurements of Hydrogen Syngas Flame Speeds at Elevated Pressures", Princeton University.
- [3] Subith Vasu, Ghazal Barari, Joseph Lopez, Luke Thompson, Bader Almansour, 2016, "Laser Ignition and Flame Speed Measurements in Oxy-Methane Mixtures Diluted with CO₂", *Journal of Energy Resources Technology*, Vol. 138, 032201-1 032201-1, pp.
- [4] Ferris, Alison M., et al. "High-Temperature Laminar Flame Speed Measurements in a Shock Tube." *Combustion and Flame*, vol. 205, 5 Apr. 2019, pp. 241–252., doi: 10.1016/j.combustflame.2019.04.007.
- [5] Beeckmann, J., et al. "Experimental Investigation of the Laminar Burning Velocities of Methanol, Ethanol, n-Propanol, and n-Butanol at High Pressure." *Fuel*, vol. 117, 2014, pp. 340–350., doi:10.1016/j.fuel.2013.09.025.
- [6] Metghalchi, M., and J.c. Keck. "Laminar Burning Velocity of Propane-Air Mixtures at High Temperature and Pressure." *Combustion and Flame*, vol. 38, 1980, pp. 143–154., doi:10.1016/0010-2180(80)90046-2.

APPENDIX

Flame Code Main

```
%% Radius/time data extraction of specific
indices (Camilie)
% initialize which indices we want to plot
id_sel_frame = (1:30);
radii = zeros(size(id_sel_frame));

for idx_r = 1:length(id_sel_frame)
    idx_frame = id_sel_frame(idx_r);
    sco2_tiff_06_contour_area_frames_radius1
        radii(1,idx_r) = radius_eff;
end

% permit figure visibility
set(0,'DefaultFigureVisible','on');

% show incremental flame growth from images
id_growth_frame = [15 16 17 18 19 20 21 22];
for idx_g = 1:length(id_growth_frame)
    idx_frame_g = id_growth_frame(idx_g);
    sco2_tiff_06_contour_area_frames_growth1
end

% collect time values from specific indices to
create
% time vector
cuttime_1 = time_s(id_sel_frame);
cuttime_s = cuttime_1.'*1;
cuttime_ms = cuttime_1.'*1000;

% define the density ratio of unburnt and
burnt gases
% density ratio changes depending on which
test file
% is being run
for filename = 'test_03.tif'
    density_ratio = 0.14939 / 1.2396;
end

for filename = 'test_01_c50.tif'
    density_ratio = .09942/.8181;
end

for filename = 'test_02_c50.tif'
    density_ratio = .30637/2.5784;
end

for filename = 'test_03_c50.tif'
    density_ratio = .14939/1.2396;
```

```
end

for filename = 'test_04_c50.tif'
    density_ratio = .09942/.8181;
end

% cm/s
    % radius length over time
    figure;
    plot(cuttime_s, radii, '-o');
    xlabel('time (s)');
    ylabel('radius (cm)');
    title('radius length over time');

    % radius vs time differentiation
    figure;
    dr_s = diff(radii)./diff(cuttime_s);
    plot(cuttime_s(2:end),dr_s, '-o');
    xlabel('time (s)');
    ylabel('change in radius (cm/s)');
    title('radius growth over time');

    % flame speed vs time
    flame_speed = dr_s*density_ratio;
    figure;
    plot(cuttime_s(2:end),flame_speed, '-o');
    xlabel('time (s)');
    ylabel('flame speed (cm/s)');
    title('flame speed at time t');

% end
```

Mask Growth Visualization

```
% part of flame main code ...
% visualizes area growth
```

```
I = frame_dat(:, :, idx_frame_g);

% initiate circle around center
% "mask" outside areas to exclude noise from
area calculation
% radius starts at b=60 for first data set
imageSize = size(I);

for filename = '_test_02_c50.tif'
    b = 60;
end

for filename = '_test_03_c50.tif'
    b = 60;
```

```

end

for filename = 'test_04_c50.tif'
    if id_growth_frame(idx_g) > 17
        b = 100;
    else b = 60;
    end
end

a = 20000;
r_mask = time_s.*a + b;

for filename = '_test_01_c50.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

for filename = '_test_02_c50.tif'
    ci = [275, 280, r_mask(idx_frame)];
end

for filename = '_test_03.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

for filename = '_test_04.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

% center and radius of circle ([c_row, c_col,
r])
[xx,yy] = ndgrid((1:imageSize(1))-
ci(1),(1:imageSize(2))-ci(2));
mask = uint8((xx.^2 + yy.^2)<ci(3)^2);
croppedImage = uint8(zeros(size(I)));
croppedImage(:, :, 1) = I(:, :, 1).*mask;

%area of center flame
th_bw = 0.1;
croppedBW = im2bw(croppedImage, th_bw);
nWhitePixels = sum(croppedBW(:));

%display figure
figure;
imshow(croppedBW);
colormap('gray');
axis equal;
axis on;
box off;
grid on;

```

```

% visualize the mask

for filename = '_test_01_c50.tif'
    ci = [275, 285];
end

for filename = '_test_02_c50.tif'
    ci = [275, 280];
end

for filename = '_test_03_c50.tif'
    ci = [275, 285];
end

for filename = '_test_04_c50.tif'
    ci = [275, 285];
end

viscircles(ci, r_mask(idx_frame_g));

Flame Code Area Calculation
% part of flame main code ...
% creates mask and calculates area of bw flame

I = frame_dat(:, :, idx_frame);

%suppress all the figures from being displayed
set(0, 'DefaultFigureVisible', 'off');

% initiate circle around center
% "mask" outside areas to exclude noise from
area calculation
% radius starts at b=60
% radius starts at b=60 for first data set
imageSize = size(I);

for filename = '_test_02c50.tif'
    b = 60;
end

for filename = '_test_03_c50.tif'
    b = 60;
end

for filename = '_test_04_c50.tif'
    if id_sel_frame(idx_r) > 17
        b = 100;
    else b = 60;
    end
end

```

```

end
    a = 20000;
    r_mask = time_s.*a + b;

for filename = '_test_01_c50.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

for filename = '_test_02_c50.tif'
    ci = [275, 280, r_mask(idx_frame)];
end

for filename = '_test_03_c50.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

for filename = '_test_04_c50.tif'
    ci = [275, 285, r_mask(idx_frame)];
end

% center and radius of circle ([c_row, c_col,
r])
    [xx,yy] = ndgrid((1:imageSize(1))-
ci(1),(1:imageSize(2))-ci(2));
    mask = uint8((xx.^2 + yy.^2)<ci(3)^2);
    croppedImage = uint8(zeros(size(I)));
    croppedImage(:, :, 1) = I(:, :, 1).*mask;

%area of center flame
th_bw = 0.1;
croppedBW = im2bw(croppedImage, th_bw);
nWhitePixels = sum(croppedBW(:));

area_cal = nWhitePixels .* (conv_px2cm.^2);

img_title_area = ['Area = ' num2str(area_cal,
'%0.3f'), ' cm^2'];

title(img_title_area, 'fontsize', 16);

timestamp = num2str(time_ms(idx_frame),
'%0.3f');
img_title_time = ['Time = ' timestamp ' ms'];

radius_eff = sqrt(area_cal/(2*pi));

fprintf(['\n' img_title_area ' at '
img_title_time ' ']);

```

```

fprintf(['\nRadius = ' num2str(radius_eff,
'%0.3f') ' cm ']);

```

Flame Code Additional Commands

% After running the code for each test, save the respective data. Then apply the following commands in the command window.

```

test_02 = load('data_test_02'); test_03 =
load('data_test_03'); test_04 =
load('data_test_04');
figure;
plot(test_02.cutttime_s(2:end),test_02.flame_sp
eed,'-o'); hold on;
plot(test_03.cutttime_s(2:end),test_03.flame_sp
eed,'-s');
plot(test_04.cutttime_s(2:end),test_04.flame_sp
eed,'-^'); hold off;
legend('2.079 atm', '1 atm', '.659 atm');
xlabel('time (s)');
ylabel('flame speed (cm/s)');
title('flame speed at time t');

```