Simulating the Effect of Thermal Conductivity on the Average Phase Velocity of Homogeneous Reaction Waves in Thin Film Self-Propagating Reactions

March 2024
12 Pages

1 Introduction

As I progress in my knowledge of chemistry, I continually learn that the chemical reactions I have been observing my entire life are much more nuanced and complex than my younger self could have ever imagined. One of my most prominent revelations as a young pyrotechnics enthusiast was the multitude of explosive and non-explosive combustion reactions being used in all corners of industry and society. From the deflagration of a fossil fuel-powered engine to the detonation of the hydrogen-oxygen mixture in a chemistry demonstration to the careful avoidance of such energetic activities in gas-powered stovetop ranges (Wikipedia contributors, Deflagration). The control of high-energy reactions like these formed a burning curiosity in my mind that I have long wished to explore. Now that I have the theoretical background from my IB chemistry courses, along with my programming skills, I can create a virtual playground of sorts to explore how I can control any reaction, real or not, to learn how they work. What I am especially interested in are energetic reactions that take the form of self-propagating waves, such as the thermite reaction (De Souza and De Lemos), and how various factors affect their properties. In particular, the effect of thermal conductivity on the reaction. While I understand that temperature almost always increases the rate of reactions, is there a point where there will be too much conduction occurring? When the system equalizes in temperature before any reaction can take place? Thermite, for example, has a relatively low thermal conductivity compared to water due to the limited surface area in the heterogeneous air-metal mixture in which it is usually contained (De Souza and De Lemos).

1.1 Background

Any reaction is governed by certain principles and values that can be measured and observed between reactions. Together, these variables form emergent properties and the complex behaviours seen in the reactions that form our industrial and scientific processes. These variables are temperature, activation energy, collision geometry, reaction orders, enthalpy, entropy and concentration (Brown and Ford). However, these factors only pertain to a microscopic point of a reaction in an extremely short timeframe. In bulk materials, which are more important to industry, macroscopic factors such as thermal conductivity, diffusion, convection, materials, porosity and types of solutes can wildly change the behaviour of a reaction (Brown and Ford).

From these variables, an infinite number of reaction behaviours may be observed. One of particular interest is reactions that propagate via wave-like structures. These occur in thermite reactions (De Souza and De Lemos) that are essential to building railways and many other structures, reaction-diffusion reactions that are of significant interest to research in not only Chemistry but Physics and Biology (Wikipedia contributors, Reaction–diffusion system), as well as the propagation of reaction in solid rocket boosters (Palečka et al.) and high explosives (Wikipedia contributors, Explosive).

The main focus of this exploration is a specific type of reaction wave called a homogeneous reaction wave. These types of reactions have been known and explored for an extremely long time as "premixed flames" in reactors that already contain all reactants for extremely exothermic (low ΔH) reactions that also have a high activation energy. One of the most prominent examples is thermite, a reaction that can reach extraordinarily high temperatures to weld steel, yet cannot be lit by any normal flame. The reactor is stable until regions neighbouring the first region that began the highly exothermic process also begin to react as the high temperatures cancel out the effect of the high activation energy of the reaction. This leads to a self-propagating reaction that takes wavelike properties (Varma et al.).

Homogeneous reaction waves (HRW) refer to a reaction wave in a reactor that has a single physical state: such as a homogeneous mixture of gases for combustion, a reaction where all reactants are dissolved in a shared solute, or where all reactants are uniformly distributed in a solid medium. This exploration studies how thermal conductivity, the rate at which the kinetic energy of the particles in a medium (thermal energy) diffuses across a medium, affects the speed of the HRW.

1.2 Research Question

How does changing the thermal conductivity of a medium (6.2, 6.4, 6.6, 6.8, 7.0, 7.2, 7.4, 7.6, 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$) affect the average phase velocity of the homogeneous reaction wave (HRW) of reaction products across a thin film self-propagating reaction?

1.3 Hypothesis

H_1: The average phase velocity of the HRW will begin to peak as the thermal conductivity increases and drop rapidly after the peak has been passed.

H_0: Thermal conductivity does not change with the average phase velocity of the HRW.

I predict the behaviour described in H_1 as I understand that higher temperatures are required to activate the high activation energy, high energy reactions used in this exploration. The mechanism of this temperature increase is the thermal conductivity. Higher thermal conductivity values increase the speed at which heat transfers through a medium (Wikipedia contributors, Thermal conductivity and resistivity). However, I predict that a peak will be reached after which there is an unstable propagation of heat in the medium that diffuses so quickly that it does not increase the temperature in any one area enough to overcome activation energy and begin a reaction.

1.4 Variables

1.4.1 Independent Variable

The thermal conductivity coefficient of the simulated thin film medium. This will be set to 6.2, 6.4, 6.6, 6.8, 7.0, 7.2, 7.4, 7.6, 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$.

1.4.2 Dependent Variable

The average phase velocity of the simulated HRW, measured in $m \cdot s^{-1}$.

1.4.3 Controlled Variables

Table 1: *Controlled Variables*

Variable	***Value***	***Units***
Initial Reactor Parameters		
Edge length of cubic grid voxel	0.01000000000000000	m
Timestep	0.5000000000000000	s
Width of reactor	100	Grid voxels (exact)
Height of reactor	100	Grid voxels (exact)
Specific heat capacity of medium	0.5000000000000000	$J \cdot g^{-1} \cdot K^{-1}$
Density of medium	4000.000000000000	$g \cdot dm^{-3}$
Diffusion constant of A and B	$1.000000000000000 \cdot 10^{-4}$	$m^2 \cdot s^{-1}$
Initial temperature	298.0000000000000	K
"Spark" temperature	9000.000000000000	K
Initial Concentration of A	100.0000000000000	$mol \cdot dm^3$
Initial Concentration of B	0.000000000000000	$mol \cdot dm^3$
Reaction Parameters of $\mathrm{A(aq)} \longrightarrow \mathrm{B(aq)} + \text{energy}$		
Enthalpy (ΔH)	−849000.0000000000	$J \cdot mol^{-1}$
Activation energy	80000.00000000000	$J \cdot mol^{-1}$
Arrhenius constant	0.001000000000000000	No units

Since this is a custom simulation lab, all variables are perfectly controlled and set to the values given to the simulation. The algorithms and techniques used in the simulation are also controlled due to the open nature of the source code and its consistency between trials. All variables in this simulation are stored as 64-bit floating-point values, meaning that all input and output variables have 16 significant digits.

All variables and units are arbitrary and found through preliminary experiments.

2 Procedure

2.1 Simulation Methodology

This exploration explores a hypothetical reaction given as:

$$\mathrm{A(aq)} \longrightarrow \mathrm{B(aq)} + \text{energy}$$

This reaction occurs in a thin film solution with variable properties such as thermal conductivity and specific heat capacity. The medium's properties do not change with temperature or the concentrations of A and B. A and B share the same diffusion constants and do not affect the heat capacity of the medium, they also do not take part in any reaction but the one above. This also occurs in an isolated system, and the total entropy of the system does not change, as the disorder never increases, and by the Second Law of Thermodynamics, can never decrease. A thin film is selected as it simplifies the simulation and allows for the easy visualization of the HRW.

For this exploration, a custom simulation (https://editor.p5js.org/hyphae/sketches/4t7sZ-eLQ) was constructed using P5.js, a programming library and hosting service that allows for digital simulations to be visualized, modified and accessed on a variety of web platforms. The simulation breaks up what would be a continuous reactor into many discrete volumes known as "voxels," short

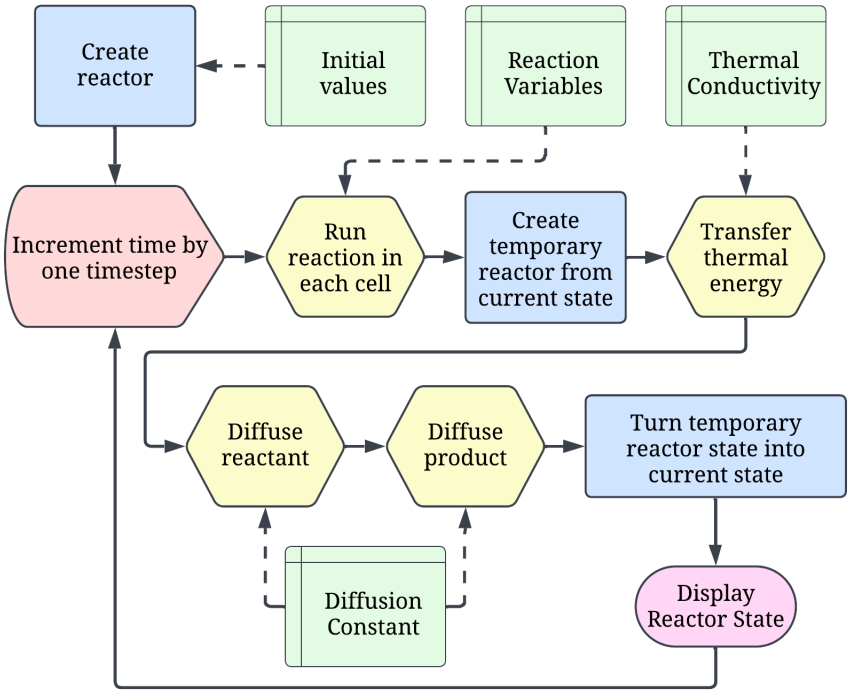
for volumetric pixels. This allows for realistic results to be achieved with minimal computational load. I created the simulation myself to have the most control over the quality and rigour of the calculations and the exploration itself. Simulations also allow extremely precise measurements of reactions that would be unreasonably difficult to create in real life, especially when changing single variables such as thermal conductivity while controlling all others.

Every relevant variable can be controlled in my simulation, including:

- The level of detail of the film (and the size of the discrete voxels)
- The amount of time that passes between each discrete time step
- The total size of the reactor
- The initial variables (temperature, concentration of reactants)
- The diffusion constant for the reactants and products
- The thermal properties of the medium assuming it changes negligibly with temperature and the concentrations of the reactants and products (the specific heat capacity and thermal conductivity constant)
- The density of the medium
- The temperature of the "spark" that begins the reaction
- The reaction variables (the enthalpy change of the reaction, the activation energy, and the Arrhenius constant of the reaction)

Each voxel in the simulation has three values assigned to it: the concentration of the reactants, the concentration of the products and the temperature of the voxel. The grid of voxels is then created and a series of algorithms are applied according to ***Figure 1***.

Figure 1: *A simplified representation of the simulation as a flowchart*



A temporary reactor grid is created to eliminate errors caused by in-place modification to the

reactor state, which leads to numerous inaccuracies and false outcomes in this class of simulations.

When simulating a reaction, a variety of formulae and models must be used to mimic the properties and behaviour of the simulation's real counterpart. For example, in a 2002 paper by Margulis et al., the foundation of their chemical simulation is a series of equations on the potential energy of the bonds begin studied. In the case of a reaction simulation, two main equations are used: Arrhenius' Law, and the Rate Law.

Arrhenius' Law describes the rate coefficient, k, of a reaction given the temperature, collision geometry (A) and activation energy (E_a) and takes the following form (Brown and Ford):

$$k = Ae^{\frac{-E_a}{RT}}$$

The other main equation, the Rate Law, takes this form (Brown and Ford):

$$r = k[A]^x[B]^y[C]^z...$$

Where r is the rate of reaction (in concentration per second), k is the rate law coefficient, [A], [B] and so on are the concentrations of the reactants, and x, y and so on are the reaction orders of the reaction.

While these equations form the base requirements of a reaction simulation, other equations must be used to model a macro-scale reactor. These equations are used only inside the simulation's code and not for analysis.

The first equation is the calorimetry equation (Brown and Ford):

$$\Delta H = \frac{mc\Delta T}{n}$$

Where ΔH is the enthalpy change of the reaction, m is the mass of the medium, c is the specific heat capacity of the medium, ΔT is the temperature change and n in the moles of reaction that occurred.

To conduct thermal energy in the simulation, the following equation is used for simple thermal conductivity in solids (Wikipedia contributors, Thermal conductivity and resistivity):

$$q = -k \cdot \frac{T_2 - T_1}{L}$$

Where q is the thermal flux, k is the thermal conductivity of the medium, T_2 and T_1 are the temperatures of the two points measured, and L is the distance between those two points. q has units of energy per area-time. However, it should be noted that while the equation used is for solid mediums and does not account for convection as would occur in liquids, in thin film reactions convection is negligible due to the thinness of the medium.

For the diffusion of the solutes in the reactor, A and B, a variation of Fick's First Law is used (Fick's laws of diffusion):

$$J = -D \cdot \frac{c_2 - c_1}{L}$$

Where J is the diffusion flux, D is the diffusivity of the solute in the medium, c_2 and c_1 are the concentrations of the solute in the two points measured, and L is the distance between those two points. J has units of concentration per area-time.

In the simulation, these formulae are used on the surrounding voxels of each voxel in the grid, with the cross-sectional area assumed to be the area of the face of a cubical voxel. Only the surrounding voxels are considered as the impact of conductivity affecting increasingly distant voxels is negligible at smaller timesteps relative to the computational load they induce.

To gather data on the speed of the phase velocity of the HRW wavefront, five voxels are selected. These voxels measure the point at which the concentration of the products reaches 90% of the original concentration of the reactants, they then record the time at which this occurred. Measuring the phase velocity of the HRW in this way eliminates the possibility of negative readings of the wavefront's presence at the analysis points. After the threshold is met, the distance of each point from the "spark" is recorded and the phase velocity of the wavefront is calculated and reported into the simulation console.

2.2 Execution

Many preliminary experiments were conducted to discover a set of initial values that created a reaction producing homogeneous reaction waves. Further experimentation pointed to the fact that a visible change in the behaviour of the reaction occurred when the thermal conductivity passed $7\ W \cdot m^{-1} \cdot K^{-1}$. This change is further described in the ***Qualitative Observation*** section. Thus, the range of this exploration's thermal conductivity values will cover this range to quantify and explore this behaviour.

The initial values of the simulation are detailed under the ***Independent Variables*** and ***Controlled Variables*** sections.

In the tests conducted for this exploration, five data collection points are used for each trial, placed at coordinates $(17, 69)$, $(32, 63)$, $(50, 50)$, $(63, 32)$ and $(69, 17)$ on the reactor grid, and will each display a value for the phase velocity of the HRW. All trials are perfectly reproducible as my simulation is accessible through: https://editor.p5js.org/hyphae/sketches/4t7sZ-eLQ

2.3 Risk Assessment

Safety: Cybersecurity best practices were rigidly adhered to for the digital portions of this exploration.

Environmental: There are no environmental concerns to consider in this exploration.

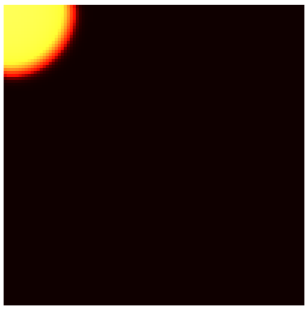
Ethical: There are no ethical concerns to consider in this exploration.

3 Observations

3.1 Qualitative Observations

As the simulation was produced in real-time, it is possible to observe the wave propagation and make qualitative observations of the more complex HRW behaviours.

Figure 2: *An example visualization of the simulation. The visualization is created by making the red channel of the image proportional to the temperature of the voxels, and the green channel proportional to the concentration of chemical B in each voxel*



From ***Figure 2***, it is possible to see that there is a gap between the temperature wave and the HRW. This appears to be a lagging effect of the temperature wave propagating faster than the HRW, and this gap increased as the thermal conductivity increased.

Another interesting property of note is that the reacted areas behind the HRW are yellow, meaning that they both have a high concentration of products and a high temperature, suggesting that diffusion and thermal conductivity do not reduce the concentration of products and the temperature of the already-reacted regions.

While the HRW quickly formed from the "spark" in the conditions of 6.2 to 7.0 $W \cdot m^{-1} \cdot K^{-1}$, between the 7.2 to 7.6 $W \cdot m^{-1} \cdot K^{-1}$ condition there is a significant period of time at the start of the simulation where the temperature appeared to disperse without the HRW forming and a stable HRW formed afterward. This "fizzling" effect became increasingly noticeable at higher thermal conductivities. At the 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$ conditions, the temperature gradient once again dispersed, but an HRW was never formed. Because an HRW never formed, the average phase velocity is considered zero.

Figure 3: *A screenshot of the 8.0 $W \cdot m^{-1} \cdot K^{-1}$ condition after the temperature gradient dispersed, showing some green voxels with high concentrations of B, and a diffuse area of red colour showing the diffused temperature gradient. Cropped for clarity.*

3.2 Quantitative Observations

Table 2: *The average phase velocity measured at points P1 through P5 in the simulation*

	Average Phase Velocity ($m \cdot s^{-1}$)		
Thermal Conductivity ($W \cdot m^{-1} \cdot K^{-1}$)	P1	P2	P3
6.2	0.00138525052666197	0.00138687268513676	0.00138920782158457
6.4	0.00139750938088022	0.00139923095658847	0.00140021144789415
6.6	0.00140719508946058	0.00140899627732239	0.00140998361153847
6.8	0.00141560462186771	0.00141605537690817	0.00141846896928093
7.0	0.00141984719316202	0.00142032489060739	0.00142275006274959
7.2	0.00141843017999519	0.00142032489060739	0.00142132016318904
7.4	0.00140719508946058	0.00140759289457605	0.00140857924539152
7.6	0.00133829288169038	0.00133827960810072	0.00134048678897923
7.8	0.00000000000000000	0.00000000000000000	0.00000000000000000
8.0	0.00000000000000000	0.00000000000000000	0.00000000000000000
	Average Phase Velocity ($m \cdot s^{-1}$)		
Thermal Conductivity ($W \cdot m^{-1} \cdot K^{-1}$)	P4	P5	
6.2	0.00138823503551508	0.00138795609409686	
6.4	0.00140061770679322	0.00140026309394599	
6.6	0.00141040246123189	0.00140998714320951	
6.8	0.00141747569323406	0.00141701599237805	
7.0	0.00142175378888768	0.00142268973008527	
7.2	0.00142175378888768	0.00142132016318904	
7.4	0.00140899627732239	0.00140858973276034	
7.6	0.00133954811957759	0.00134081796259923	
7.8	0.00000000000000000	0.00000000000000000	
8.0	0.00000000000000000	0.00000000000000000	

4 Analysis

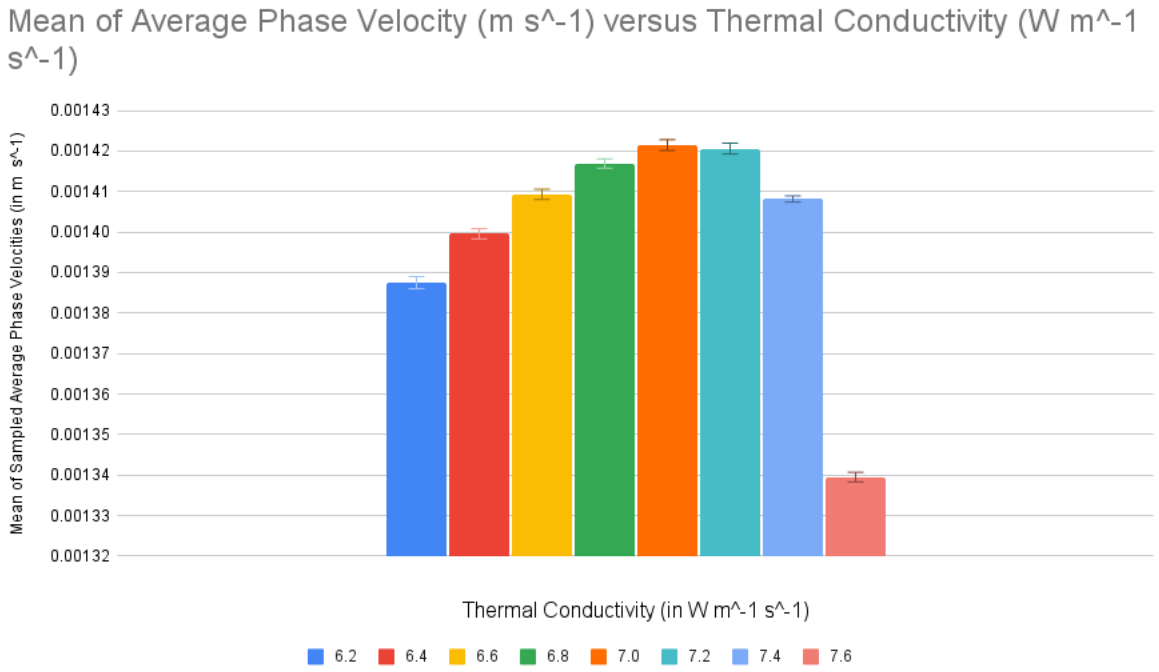
The averages and standard deviations of the phase velocity samples are collected and inserted into ***Table 3***. Note the confusing turn of phrase when taking the mean of the average phase velocity. The average phase velocity refers to the non-instantaneous measurement of the phase velocity of the HRW wavefront, and the mean refers to the statistical method of finding the average value of the measurement between sample voxels P1 through P5 in ***Table 2***. The averages and standard deviations are calculated in Google Sheets via the =AVERAGE and =STDEV functions.

Table 3: *Processed data from* ***Table 2****.*

Thermal Conductivity ($W \cdot m^{-1} \cdot K^{-1}$)	Mean of Average Phase Velocities ($m \cdot s^{-1}$)	Standard Deviation of Average Phase Velocities ($m \cdot s^{-1}$)
6.2	0.00138750443259905	0.0000015098435013951 3
6.4	0.00139956651722041	0.0000012600163775099 5
6.6	0.00140931291655257	0.0000012919919650418 1
6.8	0.00141692413073378	0.0000011396296270404 6
7.0	0.00142147313309839	0.0000013371206685899 6
7.2	0.00142062983717367	0.0000013367733148838 7
7.4	0.00140819064790218	0.0000007595658537912 5
7.6	0.00133948507218943	0.0000011893769587322 0
7.8	0.00000000000000000	0.0000000000000000000
8.0	0.00000000000000000	0.0000000000000000000

A graph may now be created to visualize the trends of the data. For clarity's sake, conditions of 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$ are not included as they disrupt the minute differences between the earlier conditions.

Figure 3: *Bar graph of the mean of the average phase velocity at each condition, visualized from the contents of* ***Table 3****. Error bars are the standard deviation of the samples. Note that the lower limit of the graph is not zero and the 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$ conditions are not included for visual clarity of the trends*



This graph clearly shows a trend in the data, with the average phase velocity increasing and then sharply decreasing after reaching a peak value. Note that after the 7.6 $W \cdot m^{-1} \cdot K^{-1}$ condition was

calculated, the average phase velocity became zero as no HRW formed in the simulation. The error bars formed by the calculation of the standard deviation between consecutive trials also do not overlap except for the 7.0 and 7.2 $W \cdot m^{-1} \cdot K^{-1}$ conditions. This implies a high level of significance of the data. However, this may be supported through the use of a One-Way Analysis of Variance (ANOVA) test under the null hypothesis (H_0) of there being no significant change in the average phase velocities between conditions, and an alternative hypothesis (H_1) of there being a significant change across the conditions. The ANOVA test is conducted using the Statistics Kingdom's ANOVA Calculator: https://www.statskingdom.com/180Anova1way.html (ANOVA Calculator - One Way ANOVA and Tukey HSD test).

ANOVA tests were performed on the raw data from ***Table 2***. The first on the entire dataset, and a second excluding the 7.8 and 8.0 $W \cdot m^{-1} \cdot K^{-1}$ conditions, in order to support that there is a significant trend in that subset of the data. The first test gave a p-value of 0, and the second a p-value of $1.110223025 \cdot 10^{-16}$. Both of these values are orders of magnitude below 0.05, therefore the results and overall trends are significant.

While the peak average phase velocity may appear to occur at 7.0 $W \cdot m^{-1} \cdot K^{-1}$, a student's T-test must be conducted to determine if the peak is significant to its neighbouring conditions 6.8 and 7.2 $W \cdot m^{-1} \cdot K^{-1}$. This test is performed using the =T.TEST() function in Google Sheets with the 2, 3 option, meaning a two-tailed heteroscedastic T-test is performed. The T-test between the 6.8 and 7.0 $W \cdot m^{-1} \cdot K^{-1}$ conditions results in a p=value of 0.00003788937668, which in under 0.05 and therefore statistically significant. The T-test between the 7.0 and 7.2 $W \cdot m^{-1} \cdot K^{-1}$ conditions results in a p=value of 0.2499236529, which is greater than 0.05 and therefore statistically insignificant. Therefore the peak value cannot be derived, and likely is produced by thermal conductivity values somewhere between 7.0 and 7.2 $W \cdot m^{-1} \cdot K^{-1}$.

5 Evaluation

5.1 Conclusion

The alternate hypothesis is very well supported by the data. The trend predicted, that the average phase velocity will increase to a peak and steeply drop off as thermal conductivity increases. The trend is very obvious in the data and is supported by very low p-values when descriptive statistical methods are employed to determine the strength of the data. The data supporting the hypothesis is overall extremely strong. The intuition behind my alternate hypothesis is also supported by the "fizzling" effect observed in trials occurring past the peak average phase velocity, as the temperature wave appeared to exceed the HRW and became more diffuse as the thermal conductivity increased. The answer to the research question in **Section 1.2** is that the average phase velocity of the HRW increases with the thermal conductivity up to a point, at which it peaks and steeply drops to zero as an HRW fails to form as thermal conductivity continues to increase.

While it is impossible to recreate the exact method of the simulation in real life, the trends and behaviours of the simulation create a set of behaviours that can be expected in a real experiment and hence allow further research into the phenomena created when thermal conductivity is altered in a thin film self-propagating homogeneous reaction wave.

5.2 Strengths

This exploration has numerous strengths that come from its nature as a simulation. A simulation allows for the extremely precise control of input variables to a degree impossible in conventional experiments. It allows unparalleled accuracy due to the way that computations are performed in a computer and the fact that there is no uncertainty due to the level of control. This simulation is also very versatile, as it is possible to simulate nearly any simple thin film reaction using this framework.

5.3 Limitations

The main limitation of this exploration is the fact that it solely consists of a simulation without real experiments to benchmark the parity of the simulation to real values. This is a source of random error as it is impossible to predict the effect that this potential error has on the data, and may overestimate some values and underestimate others depending on the input. While the simulation currently attempts to minimize this error by using equations and methods that are grounded in experimental evidence, future attempts to benchmark and tune the simulation would further minimize this error. If this simulation were to be calibrated to real life, it would be possible to accurately predict the outcomes and behaviours of many types of reactions with applications in chemical engineering and industry.

Another limitation of this exploration is the method of measuring the average phase velocity of the HRW. The average phase velocity over a relatively long distance is computed, while a far more accurate method would be to measure the near-instantaneous phase velocity over numerous smaller periods and take the mean of those values. This is a source of systematic error as it also fails to account for the delayed start observed during some conditions as well as the potential acceleration of the HRW wavefront. The aforementioned method of measuring the phase velocity over smaller distances would minimize this source of error and allow for the verification of uniform phase velocity on the wavefront.

The final source of error in this exploration is the fact that this is a discrete simulation. This is a source of random error as reactions do not occur in real life in discrete voxels over discrete timesteps, thus leading to a reduction in the accuracy of the simulation. While it is impossible to create a continuous simulation without the use of quantum computing, this source of error may be minimized by reducing the timesteps and discrete length units significantly, as well as increasing the area by which the diffusion and thermal conductivity equations have an effect.

5.4 Further Exploration

This exploration opens up avenues of further research into this situation and exploration topic, beyond the improvements already proposed. The simulation can become three-dimensional and can then be applied to a wider range of applications in bulk materials. Another area for exploration is using multiple “sparks” to create complex reaction behaviour, possibly shaping a reaction such that the reaction follows a particular pattern, such as a shaped charge. Or, the medium could be altered to become a heterogeneous mixture to achieve a similar effect. The simulation created allows for simple changes to be made to create powerful new applications that extend far beyond the scope of this exploration.

6

Works Cited

"ANOVA Calculator - One Way ANOVA and Tukey HSD test", www . statskingdom . com / 180Anova1way.html.

Brown, Catrin, and Mike Ford. *Pearson Baccalaureate Chemistry Higher Level 2nd Edition Print and Online Edition for the IB diploma*. Prentice Hall, 1 Dec. 2008.

De Souza, Kesiany M., and Marcelo J. S. De Lemos. "Advanced one-dimensional modeling of thermite reaction for thermal plug and abandonment of oil wells". *International Journal of Heat and Mass Transfer*, vol. 205, 1 May 2023, p. 123913. https://doi.org/10.1016/j.ijheatmasstransfer.2023.123913.

Margulis, Claudio J., et al. "Computer simulation of a "Green Chemistry" Room-Temperature ionic solvent". *The Journal of Physical Chemistry B*, vol. 106, no. 46, 26 Oct. 2002, pp. 12017–21. https://doi.org/10.1021/jp021392u.

Palečka, Jan, et al. "Percolating Reaction–Diffusion Waves (PERWAVES)—Sounding rocket combustion experiments". *Acta Astronautica*, vol. 177, 1 Dec. 2020, pp. 639–51. https://doi.org/10.1016/j.actaastro.2020.07.033.

Varma, Arvind, et al. "Complex behavior of self-propagating reaction waves in heterogeneous media". *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 19, 15 Sept. 1998, pp. 11053–58. https://doi.org/10.1073/pnas.95.19.11053.

Wikipedia contributors. "Deflagration", 21 Mar. 2024, en.wikipedia.org/wiki/Deflagration.

———. "Explosive", 25 Mar. 2024, en.wikipedia.org/wiki/Explosive.

———. "Fick's laws of diffusion", 13 Mar. 2024, en.wikipedia.org/wiki/Fick%27s_laws_of_diffusion.

———. "Reaction–diffusion system", 17 Nov. 2023, en.wikipedia.org/wiki/Reaction%E2%80%93diffusion_system.

———. "Thermal conductivity and resistivity", 25 Mar. 2024, en.wikipedia.org/wiki/Thermal_conductivity_and_resistivity.

7 Appendix

7.1 Simulation Code

Link to code: https://editor.p5js.org/hyphae/sketches/4t7sZ-eLQ

```
//SIMULATION PARAMETERS: Change these values to change the simulation settings
//Note that all variables in this simulation are stored as 64-bit
↪ floating-point values, meaning that all variables have 16 significant
↪ digits in the background of the simulation, regardless of the significance
↪ of the input variables

//Thermal conductivity of medium (in W m^-1 K-1)
thermConductivity = 6.2;

// Length step (voxel edge length) (1 = 1m, 0.001=1mm)
steplength = 0.01;

//time step (amount of modelled time that passes between each "frame" of the
↪ simulation) in seconds
t = 0.5;

// Size of the reactor (width, height) multiples of the length step, or units
↪ (referred to as voxels)
// Notes that depth is fixed to a singular voxel to best model a 2-dimensional
↪ reactor
reactorWidth = 100;
reactorHeight = 100;

// volume of voxel is the length step cubed (m^3) times 1000 (dm^3)
volume = steplength * steplength * steplength * 1000;

//specific heat capacity of medium (in J g-1 K-1)
specheatcapacity = 0.5;

//density of medium per 1dm^3 in g dm^-3
density = 4000;

//specific heat capacity of voxel (the number of Joules of energy it takes to
↪ raise the temperature of a voxel by 1 Kelvin)
voxelheat = volume * density * specheatcapacity;

//diffusion constant of reactants and products (in m^2 s^-1)
diffusion = 1 * 10 ** -4;

//enthalpy of reaction (in J mol^-1)
dH = -849000;
```

```
// activation energy of reaction (in J mol^-1)
Ea = 80000;

//Arrhenius factor

a = 0.001;

//inital temperature of reactor (in K)
initTemp = 298;

//initial tmeperature of "spark"
sparktemp = 9000;

//inital concentration of reactants (in mol dm^3)
initconc = 100;

//display simulation?
displaysim = true;

//Reaction "completion" threshold (in amount of product made relative to inital
↪  amount of reactant)
thresh = 0.9;
//Internal variables, do not modify
let rttwo;
thermfluxpre =
  (steplength * steplength * (thermConductivity / steplength) * t) / voxelheat;
thermfluxprert =
  (steplength * steplength * (thermConductivity / (rttwo * steplength)) * t) /
  voxelheat;

diffluxpre = steplength * steplength * (diffusion / steplength) * t;
diffluxprert =
  steplength * steplength * (diffusion / (rttwo * steplength)) * t;

let reactor;
let tempreactor;
let time = 0;
let done = false;
t1 = false;
t2 = false;
t3 = false;
t4 = false;
t5 = false;

class voxel {
  constructor(conc, temp) {
```

```
    this.conc = conc;
    this.temp = temp;
    this.pro = 0;
    this.k = 0;
  }

  react() {
    this.k = a * exp((-1 * Ea) / (this.temp * 8.31));
    let concdiff = this.conc * this.conc * this.k * t;
    if (concdiff > this.conc) {
      concdiff = this.conc;
    }
    this.conc -= concdiff;
    this.pro += concdiff;
    let molmade = concdiff * volume;
    this.temp += (-1 * dH * molmade) / voxelheat;
  }
}

function makeReactor(cols, rows, conc, temp) {
  let arr = new Array(cols);
  for (let i = 0; i < arr.length; i++) {
    arr[i] = new Array(rows);
    // Fill the array with class objects
    for (let j = 0; j < arr[i].length; j++) {
      arr[i][j] = new voxel(conc, temp);
    }
  }
  return arr;
}

function setup() {
  rttwo = sqrt(2);
  thermfluxpre = steplength * steplength * (thermConductivity / steplength) * t;
  thermfluxprert =
    steplength * steplength * (thermConductivity / (rttwo * steplength)) * t;

  difffluxpre = steplength * steplength * (diffusion / steplength) * t;
  diffluxprert =
    steplength * steplength * (diffusion / (rttwo * steplength)) * t;

  createCanvas(5 * reactorWidth, 5 * reactorHeight);

  reactor = makeReactor(reactorWidth, reactorHeight, initconc, initTemp);
  tempreactor = Array.from(reactor);

  reactor[0][0].temp = sparktemp;
```

```
}

// Check if a coordinate is within the bounds
function withinGrid(x, y) {
  return x >= 0 && x <= reactorWidth - 1 && y >= 0 && y <= reactorHeight - 1;
}

function tempConduct(x, y) {
  let temp = reactor[x][y].temp;

  if (withinGrid(x - 1, y - 1)) {
    temp += thermfluxprert * (reactor[x - 1][y - 1].temp - reactor[x][y].temp);
  }
  if (withinGrid(x + 1, y - 1)) {
    temp += thermfluxprert * (reactor[x + 1][y - 1].temp - reactor[x][y].temp);
  }
  if (withinGrid(x - 1, y + 1)) {
    temp += thermfluxprert * (reactor[x - 1][y + 1].temp - reactor[x][y].temp);
  }
  if (withinGrid(x + 1, y + 1)) {
    temp += thermfluxprert * (reactor[x + 1][y + 1].temp - reactor[x][y].temp);
  }

  if (withinGrid(x - 1, y)) {
    temp += thermfluxpre * (reactor[x - 1][y].temp - reactor[x][y].temp);
  }
  if (withinGrid(x + 1, y)) {
    temp += thermfluxpre * (reactor[x + 1][y].temp - reactor[x][y].temp);
  }
  if (withinGrid(x, y + 1)) {
    temp += thermfluxpre * (reactor[x][y + 1].temp - reactor[x][y].temp);
  }
  if (withinGrid(x, y - 1)) {
    temp += thermfluxpre * (reactor[x][y - 1].temp - reactor[x][y].temp);
  }
  return temp;
}

function reactDiffuse(x, y) {
  temp = reactor[x][y].conc;

  if (withinGrid(x - 1, y - 1)) {
    temp += difffluxprert * (reactor[x - 1][y - 1].conc - reactor[x][y].conc);
  }
  if (withinGrid(x + 1, y - 1)) {
    temp += difffluxprert * (reactor[x + 1][y - 1].conc - reactor[x][y].conc);
  }
```

```
  if (withinGrid(x - 1, y + 1)) {
    temp += difffluxprert * (reactor[x - 1][y + 1].conc - reactor[x][y].conc);
  }
  if (withinGrid(x + 1, y + 1)) {
    temp += difffluxprert * (reactor[x + 1][y + 1].conc - reactor[x][y].conc);
  }

  if (withinGrid(x - 1, y)) {
    temp += difffluxpre * (reactor[x - 1][y].conc - reactor[x][y].conc);
  }
  if (withinGrid(x + 1, y)) {
    temp += difffluxpre * (reactor[x + 1][y].conc - reactor[x][y].conc);
  }
  if (withinGrid(x, y + 1)) {
    temp += difffluxpre * (reactor[x][y + 1].conc - reactor[x][y].conc);
  }
  if (withinGrid(x, y - 1)) {
    temp += difffluxpre * (reactor[x][y - 1].conc - reactor[x][y].conc);
  }
  return temp;
}

function proDiffuse(x, y) {
  temp = reactor[x][y].pro;

  if (withinGrid(x - 1, y - 1)) {
    temp += difffluxprert * (reactor[x - 1][y - 1].pro - reactor[x][y].pro);
  }
  if (withinGrid(x + 1, y - 1)) {
    temp += difffluxprert * (reactor[x + 1][y - 1].pro - reactor[x][y].pro);
  }
  if (withinGrid(x - 1, y + 1)) {
    temp += difffluxprert * (reactor[x - 1][y + 1].pro - reactor[x][y].pro);
  }
  if (withinGrid(x + 1, y + 1)) {
    temp += difffluxprert * (reactor[x + 1][y + 1].pro - reactor[x][y].pro);
  }

  if (withinGrid(x - 1, y)) {
    temp += difffluxpre * (reactor[x - 1][y].pro - reactor[x][y].pro);
  }
  if (withinGrid(x + 1, y)) {
    temp += difffluxpre * (reactor[x + 1][y].pro - reactor[x][y].pro);
  }
  if (withinGrid(x, y + 1)) {
    temp += difffluxpre * (reactor[x][y + 1].pro - reactor[x][y].pro);
  }
```

```
  if (withinGrid(x, y - 1)) {
    temp += diffluxpre * (reactor[x][y - 1].pro - reactor[x][y].pro);
  }
  return temp;
}

function draw() {
  background(0);
  noStroke();
  for (let x = 0; x <= reactorWidth - 1; x++) {
    for (let y = 0; y <= reactorHeight - 1; y++) {
      reactor[x][y].react();
    }
  }

  for (let x = 0; x <= reactorWidth - 1; x++) {
    for (let y = 0; y <= reactorHeight - 1; y++) {
      tempreactor[x][y].temp = tempConduct(x, y);
    }
  }

  reactor = Array.from(tempreactor);

  if (displaysim) {
    for (let x = 0; x <= reactorWidth - 1; x++) {
      for (let y = 0; y <= reactorHeight - 1; y++) {
        fill(reactor[x][y].temp / 20, reactor[x][y].pro * 5, reactor[x][y].pro);
        square(x * 5, y * 5, 5);
      }
    }
    time += t;
    if (reactor[17][69].pro >= initconc * thresh && !t1) {
      print("P1 speed is: " + (sqrt(17 * 17 + 69 * 69) * steplength) / time);
      t1 = true;
    }
    if (reactor[32][63].pro >= initconc * thresh && !t2) {
      print("P2 speed is: " + (sqrt(32 * 32 + 63 * 63) * steplength) / time);
      t2 = true;
    }
    if (reactor[50][50].pro >= initconc * thresh && !t3) {
      print("P3 speed is: " + (sqrt(50 * 50 * 2) * steplength) / time);
      t3 = true;
    }
    if (reactor[63][32].pro >= initconc * thresh && !t4) {
      print("P4 speed is: " + (sqrt(32 * 32 + 63 * 63) * steplength) / time);
      t4 = true;
    }
```

```
    if (reactor[69][17].pro >= initconc * thresh && !t5) {
      print("P5 speed is: " + (sqrt(17 * 17 + 69 * 69) * steplength) / time);
      t5 = true;
    }
    if (reactor[reactorWidth - 1][reactorHeight - 1].pro >= initconc * 0.5) {
      noLoop();
    }
  }
}
```