Journal of Applied Engineering Mathematics

Volume 10, December 2023

ROTATING DETONATION ENGINE MODELING

Noah Cahill and Joshua Taylor

Mechanical Engineering Department
Brigham Young University
Provo, Utah 84602
jct418@byu.edu, ncahill@byu.edu

ABSTRACT

Rotating Detonation Engines are experimental combustion engines being researched for their projected improvements in thermodynamic cycle efficiency. They facilitate rapid combustion by rotating shocks. This paper proposes an analytical method derived from the Heat Diffusion Equation using integral transform methods. A 1-D and 2-D solution are developed to model the transient temperature field inside an annular chamber during combustion and examine the effects different parameters have on heating inside the chamber.

NOMENCLATURE

T = Temperature distribution
θ = Angular coordinate
t = Time
r = Radial coordinate
k = Conduction coefficient
S_0 = Combustion source power
t_b = Burn time
ω = Flame front angular speed
ρ = Density
c_p = Specific heat
α = Thermal diffusivity

INTRODUCTION

Rotating Detonation Engines offer a primary advantage over traditional combustion engines in that they can sustain long duration burns with less fuel. Though these engines have shown experimental promise, quantifying characteristics including flame propagation in these engines has proven to be extraordinarily difficult.

Rotating Detonation Engines operate on the working principle that fuel is axially fed through an annular chamber with pre-fed oxidizer. At engine detonation, a strong shock immediately traverses the annular chamber providing the necessary pressure and temperature rise to facilitate combustion reactions between the fuel and oxidizer. Though shocks are modeled as adiabatic, when coupled with a detonation event, they dissipate energy through the combustion chamber.

$$1)\quad \frac{1}{r}\frac{\partial^2 T}{\partial \theta^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T}{\partial r}\right)+\ k\frac{\partial^2 T}{\partial z^2}+\dot{q}=\frac{\rho c_p}{k}\frac{\partial T}{\partial t}$$

The purpose of this study is to visualize the temperature response of the combustion chamber due to a moving shock. Standard in most heat transfer problems, the Heat Diffusion Equation (1) is used to predict temperatures and temperature gradients in any domain. Though modeling would require the simultaneous solution of other conservation equations, the model is simplified by ignoring coupled equations. This simplification would allow the possibility of obtaining a closed-form analytical solution for visualization.

MODEL

Formulation 1-D

To begin modeling the system, the 1-D heat equation in cylindrical coordinates (1) was simplified. Because Rotating Detonation Engines are thin and insulated at the walls, temperature dependence with radius and height was neglected.

Boundary conditions were developed so that the interface points between 0 radians and 2π radians would have the same temperature and temperature gradient (2) (3).

$$2)\quad T(\theta,t)=T(\theta+2\pi,t)$$

$$3)\quad \frac{\partial T}{\partial \theta}(\theta,t)=\frac{\partial T}{\partial \theta}(\theta+2\pi,t)$$

$$4)\quad T(\theta,0)=T_\infty-T_\infty=\ 0$$

The shock wave is represented as a moving heat source in the tangential direction by angular rate ω. To simulate the effect of combustion duration, an energy decay was implemented by way of the decaying exponential $e^{-\frac{3t}{t_b}}$ in the source term (5) (6).

$$5)\qquad \frac{k}{r_1}\frac{\partial^2 T}{\partial\theta^2} + S_0 e^{-\frac{3t}{t_b}}\delta(\theta-\omega t) = \rho c_p \frac{\partial T}{\partial t}$$

$$6)\qquad \frac{\alpha}{r_1}\frac{\partial^2 T}{\partial\theta^2} + \frac{S_0}{\rho c_p} e^{-\frac{3t}{t_b}}\delta(\theta-\omega t) = \frac{\partial T}{\partial t}$$

Solution 1-D

Circular Transform

$$7)\qquad \Phi\left\{\frac{\alpha}{r_1}\frac{\partial^2 T}{\partial\theta^2} + \frac{S_0}{\rho c_p} e^{-\frac{3t}{t_b}}\delta(\theta-\omega t) = \frac{\partial T}{\partial t}\right\}$$

$$8)\qquad \frac{\alpha}{r_1}(-n^2\overline{T}) + \frac{S_0}{\rho c_p} e^{-\frac{3t}{t_b}}\cos\left(n(\theta-\omega t)\right) = \frac{\partial\overline{T}}{\partial t}$$

Laplace Transform

$$9)\qquad \mathcal{L}\left\{\frac{\alpha}{r_1}(-n^2\overline{T}) + \frac{S_0}{\rho c_p} e^{-\frac{3t}{t_b}}\cos\left(n(\theta-\omega t)\right) = \frac{\partial\overline{T}}{\partial t}\right\}$$

$$10)\qquad \frac{\alpha}{r_1}(-n^2\overline{\overline{T}}) + \frac{S_0}{\rho c_p}\left[\frac{\cos(n\theta)\left(s+\frac{3}{t_b}\right)+n\omega\sin(n\theta)}{\left(s+\frac{3}{t_b}\right)+n^2\omega^2}\right] = s\overline{\overline{T}}$$

After both transformations have been made, the equation is solved for the transformed temperature variable. The resulting equation (11) can then be input into MATLAB to invert each of the transformations and be plotted.

$$11)\qquad \overline{\overline{T}} = \frac{S_0}{\rho c_p\left(s+n^2\frac{\alpha}{r_1}\right)}\left[\frac{\cos(n\theta)\left(s+\frac{3}{t_b}\right)+n\omega\sin(n\theta)}{\left(s+\frac{3}{t_b}\right)+n^2\omega^2}\right]$$

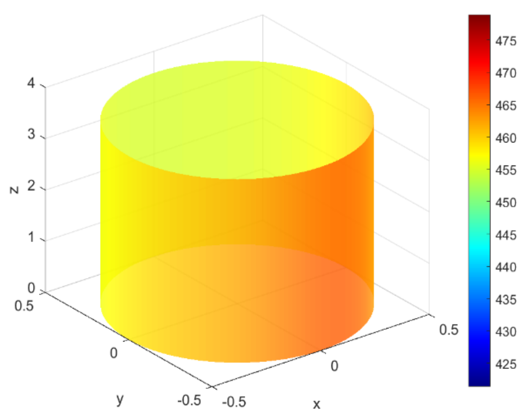

Figure 1. Temperature distribution in the combustion chamber after 11.82 seconds, using 1D model.

Formulation 2-D

The 2-D system derivation was not all that much different from the 1-D derivation. The 2-D heat equation in cylindrical coordinates was taken and simplified (1). Temperature was assumed to not depend on radius.

$$12)\qquad T(\theta,z,t) = T(\theta+2\pi,z,t)$$

$$13)\qquad \frac{\partial T}{\partial\theta}(\theta,z,t) = \frac{\partial T}{\partial\theta}(\theta+2\pi,z,t)$$

$$14)\qquad k\frac{\partial T}{\partial z}(\theta,L,t) + hT(\theta,L,t) = 0$$

Different from the 1-D formulation, an axial Robin boundary condition was used to model the convection out of the top (z = L) of the engine (14). Two separate boundary conditions were formulated for the base (z = 0) of the engine: Neumann (15) and Dirichlet (16).

$$15)\qquad \frac{\partial T}{\partial z}(\theta,0,t) = 0$$

$$16)\qquad \mathrm{T}(\theta,L,t) = T_\infty - T_\infty = 0$$

The shock wave is represented as a moving heat source with some wave angle ε that varies helically with height (17). This would be representative of an oblique shock that traverses the chamber during combustion.

$$17)\qquad \frac{1}{r_1}\frac{\partial^2 T}{\partial\theta^2} + \frac{\partial^2 T}{\partial z^2} + S_0 e^{-\frac{3t}{t_b}}\delta(\theta-\omega t+\varepsilon z) = \frac{1}{\alpha}\frac{\partial T}{\partial t}$$

Solution 2-D

Circular Transform

$$18)\qquad \Phi\left\{\frac{1}{r_1}\frac{\partial^2 T}{\partial\theta^2} + \frac{\partial^2 T}{\partial z^2} + \alpha S_0 e^{-\frac{3t}{t_b}}\delta(\theta-\omega t+\varepsilon z) = \frac{\partial T}{\partial t}\right\}$$

$$\rightarrow -\frac{n^2\alpha}{r_1}\overline{T} + \frac{\partial^2\overline{T}}{\partial z^2} + \alpha S_0 e^{-\frac{3t}{t_b}}\cos\left(n(\theta-\omega t+\varepsilon z)\right) = \frac{\partial\overline{T}}{\partial t}$$

Fourier Transform

$$19)\qquad \mathcal{F}\left\{-\frac{n^2\alpha}{r_1}\overline{T} + \frac{\partial^2\overline{T}}{\partial z^2} + \alpha S_0 e^{-\frac{3t}{t_b}}\cos\left(n(\theta-\omega t+\varepsilon z)\right) = \frac{\partial\overline{T}}{\partial t}\right\} \rightarrow -\frac{n^2\alpha}{r_1}\overline{\overline{T}} - \alpha{\mu_n}^2\overline{\overline{T}} + \alpha S_0 e^{-\frac{3t}{t_b}}\int_0^L \cos(\mu_n z)\cos\left(n(\theta-\omega t+\varepsilon z)\right)dz = \frac{\partial\overline{\overline{T}}}{\partial t}$$

Dirichlet-Robin BC

$$20)\quad Z_n = \sin(\mu_n z)$$

$$21)\quad \mu_n \cos(\mu_n L) + \frac{h}{k}\sin(\mu_n L) = 0$$

$$22)\quad \|Z_n\|^2 = \frac{L}{2} - \frac{\sin(2\mu_n L)}{4\mu_n}$$

Neumann-Robin BC

$$23)\quad Z_n = \cos(\mu_n z)$$

$$24)\quad \mu_n \sin(\mu_n L) - \frac{h}{k}\cos(\mu_n L) = 0$$

$$25)\quad \|Z_n\|^2 = \frac{L}{2} + \frac{\sin(2\mu_n L)}{4\mu_n}$$

Laplace Transform

$$26)\quad \mathcal{L}\left\{-\frac{n^2\alpha}{r_1}\bar{\bar{T}} - \alpha\mu_n{}^2\bar{\bar{T}} + \alpha S_0 e^{-\frac{3t}{t_b}}\int_0^L \cos(\mu_n z)\cos\big(n(\theta - \omega t + \varepsilon z)\big)\,dz = \frac{\partial\bar{\bar{T}}}{\partial t}\right\} \rightarrow -\frac{n^2\alpha}{r_1}\hat{\bar{\bar{T}}} - \alpha\mu_n{}^2\hat{\bar{\bar{T}}} + \mathcal{L}\left\{\alpha S_0 e^{-\frac{3t}{t_b}}\int_0^L \cos(\mu_n z)\cos\big(n(\theta - \omega t + \varepsilon z)\big)\,dz\right\} = s\hat{\bar{\bar{T}}}$$

After both transformations have been made, the equation is solved for the transformed temperature variable. The resulting equation (27) can then be input into MATLAB to invert each of the transformations and be plotted.

$$26)\quad \hat{\bar{\bar{T}}} = \frac{\mathcal{L}\left\{\alpha S_0 e^{-\frac{3t}{t_b}}\int_0^L \cos(\mu_n z)\cos\big(n(\theta-\omega t+\varepsilon z)\big)dz\right\}}{s + \frac{n^2\alpha}{r_1} + \alpha\mu_n{}^2}$$

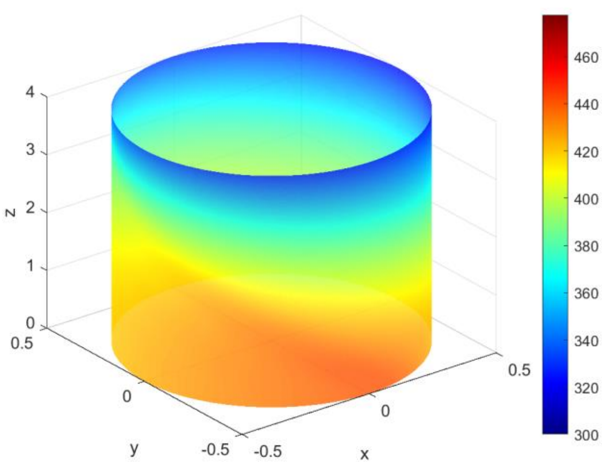

Figure 2. Temperature distribution in the combustion chamber after 11.82 seconds, using a Neumann boundary condition at z = 0.

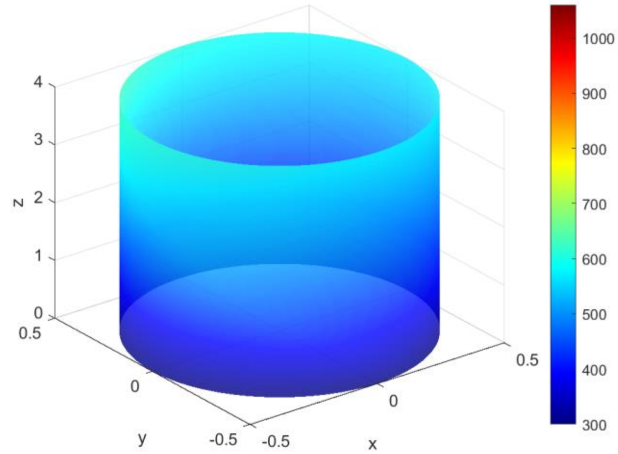

Figure 3. Temperature distribution in the combustion chamber after 11.82 seconds, using a Dirichlet boundary condition at z = 0.

RESULTS

1-D

After 11.8 seconds, the temperature distribution in the combustion chamber began to homogenize (see Figure 1). For simplified, macroscopic applications, such a model could be useful to model the temperature within the combustion chamber of a rotating detonation engine. This model could also be useful if only a model of a thin cross section of the combustion chamber is required.

2-D

After a burn duration of 11.82 seconds, Figure 3 showed the greatest increase in average temperature over the upper half of the annular domain. Meanwhile, Figure 2 showed the greatest increase in average temperature over the lower half of the annular domain. Both figures visually show noticeably greater heat transfer in the axial direction over the rotational direction. However, when the aspect ratio of the figure is considered, the rate of heat transfer in the axial direction is deceptively higher. Temperature striations in Figure 2 indicate net heat transfer from the bottom of the chamber to the top of the chamber. Temperature striations in Figure 3 indicate net heat transfer from the top of the chamber to the bottom of the chamber. These differences were expected since the bottom boundary was insulated in Figure 2 and isothermal in Figure 3.

CONCLUSIONS

As can be seen by the results obtained above, a simplified heat transfer problem can be formulated for a Rotating Detonation Engine. The two-dimensional model allowed for the variation of the wave angle and the visualization of the heat gradient in both the rotational and axial directions. The solutions, as presented in the figures, are dependent on boundary conditions in the cylindrical domain. They are standalone and represent the best attempt of the authors to obtain closed form models using integral transform methods. On admission, developed solutions in this text do not consider the coupled nature of conservation laws. For more accurate modeling, numerical methods would need to be employed to visualize the coupled nature of heat and mass transfer.

REFERENCES

[1] Sousa, J., Braun, J., & Paniagua, G. (2021). Development of a fast evaluation tool for rotating detonation combustors. *Applied Mathematical Modelling*, *52*. https://doi.org/10.1016/j.apm.2017.07.019

APPENDIX A

1-D MATLAB Code

```
%% Setup
dim = 100;
tdim = 100;
a = 1;
pcp = 1;
s0 = 100;
T_in = 300; %K
tb = 30; %s burn time
sig = 3/tb; %decay exponent
rot = 2; %rotations of combusion
om = rot*2*pi/tb; %rad/s
r1 = 0.5; %m
nf = 5; %fourier term count

%% Inverse integral tranformations
syms s n th t;
T_il =
(s0/(pcp*(s+n^2*a/r1)))*((cos(n*th)*(s+sig)+n*
om*sin(n*th))/((s+sig)^2+n^2*om^2));
T_i = ilaplace(T_il);
T_i0 = (s0/(pcp*sig))*(1-exp(-t/sig));
T_isum = symsum(T_i,n,1,nf);
T = (1/(2*pi))*T_i0+(1/pi)*T_isum;
thl = linspace(0,2*pi,dim);
tl = linspace(0,tb,tdim);
r = linspace(0,r1,dim);

T_out = zeros(length(thl),length(tl));
T_out = T_out +T_in;
f = waitbar(0,"Computing solution...");
for thi = 1:length(thl)
    th = thl(thi);
    waitbar(thi/length(thl));
    for ti = 1:length(tl)
        waitbar(thi/length(thl),f,"Computing
solution...("+ti+"/"+length(tl)+")");
        t = tl(ti);
        T_out(thi,ti) = T_out(thi,ti) +
double(subs(T));
    end
end
close(f);

%% Plotting

figure(1);
light
lighting gouraud
for ii = 1:length(tl)

clf;
   T_mat = zeros(length(thl),length(r)) +
T_in;
    T_mat(:,length(r)) = T_out(:,ii);
    T_mat(:,1:end-1) = NaN;
    [x,y,z] = pol2cart(thl,r1,T_mat);
    plot3(x,y,z);
    zlim([T_in,max(max(T_out))])
    pause(0.1)
end

%% Cylindrical Plot
figure(2)
[X,Y,Height] = cylinder(r1,dim-1);
[angle,radius] = cart2pol(X,Y);
L = max(max(T_out))-T_in;
Height = Height*L;

s = surf(X,Y,Height/50,[T_out(:,1)';
T_out(:,1)'], 'FaceAlpha', 0.8, 'EdgeColor',
'none');
colorbar;
clim([0.88*max(max(T_out)),max(max(T_out))])
colormap(jet);
s.FaceColor = 'interp';
for ii = 1:length(tl)
    s.CData = [T_out(:,ii)'; T_out(:,ii)'];
    pause(0.15)
end
colorbar;

%% Static Cylindrical Plot
figure(3)
[X,Y,Height] = cylinder(r1,dim-1);
[angle,radius] = cart2pol(X,Y);
L = max(max(T_out))-T_in;
Height = Height*L;

s = surf(X,Y,Height/50,[T_out(:,40)';
T_out(:,40)'], 'FaceAlpha', 0.8, 'EdgeColor',
'none');
xlabel("x");
ylabel("y");
zlabel("z");
colorbar;
clim([0.88*max(max(T_out)),max(max(T_out))])
colormap(jet);
s.FaceColor = 'interp';
colorbar;
```


APPENDIX B

2-D MATLAB Code

```
%% Setup
syms z theta omega t m
L = 4;
R = 0.5;
H = 2;
S0 = 100;
alpha = 1;
lambda = 0:0.01:20;

% Eigenvalue problem for different BC

%val = -lambda.*sin(lambda*L) +
H*cos(lambda*L);
val =   lambda.*cos(lambda*L) +
H*sin(lambda*L);

% Roots of Eigenvalue problem
j = 1;
for i = 2:length(lambda)
    if val(i) == 0
        lamn(j) = lambda(i);
        j = j + 1;
    elseif val(i)/abs(val(i)) ~= val(i-
1)/abs(val(i-1))
        lamn(j) = (lambda(i) + lambda(i-1))/2;
        j = j + 1;
    else

    end
end

syms s

% Eigenvalue functions for different BC

%Zn = cos(lamn*z);
Zn = sin(lamn*z);

%Nn = (L/2 + sin(2*lamn*L)./(4*lamn));
Nn = (L/2 - sin(2*lamn*L)./(4*lamn));


source = S0*alpha*cos(m*(theta-
omega*t+0.7*z));

source_fourier = int(Zn*source,z,[0 L]);
T_laplace = laplace(source_fourier)./(s +
m^2*alpha/R + alpha*lamn.^2);

T_fourier_circular = ilaplace(T_laplace,s,t);
T_circular = sum(T_fourier_circular.*Zn./Nn);
T_circular_0 = subs(T_circular,m,0);

T_circular_m = subs(T_circular,m,1:1:10);

N = 15;
T = T_circular_0/(2*pi) +
sum(T_circular_m)/pi;
T = subs(T,omega,2*pi/N);
matlabFunction(T,"File","t_ret");

%% Function Iteration
dim = 100;
tdim = 100;
T_in = 300;
tb = 30;
r1 = R;
thl = linspace(0,2*pi,dim);
zl = linspace(0,L,dim);
tl = linspace(0,tb,tdim);

r = linspace(0,r1,dim);

T_out =
zeros(length(thl),length(zl),length(tl));
T_out = T_out +T_in;
f = waitbar(0,"Computing solution...");
for thi = 1:length(thl)
    for zi = 1:length(zl)
        T_out(thi,zi,ti) = T_out(thi,zi,ti) +
        t_ret(tl(ti),thl(thi),zl(zi));
        end
    end
end

close(f);
```

