Geometric Models of Sintering

By

Dr. David Keffer

MSE 506
Department of Material Science & Engineering
University of Tennessee
Knoxville, TN 37996-2100
First Written: October, 2007
Last Updated: April, 2014

Table of Contents

I. Purpose:

The objective of this project is to develop a geometric model for the sintering process and to investigate the effect of geometry on sintering dynamics. In this first stage of the work, we adopt a standard model of sintering from the literature and examine the dynamics for two-dimensional systems as a function of the size ratio of the two particles being sintered.

II. Description of Physical System:

The process of sintering describes the aggregration of two or more particles in order to minimize the free energy of the system. Typically there is a surface energy penalty such that this minimization of free energy can be described as a minimization of surface area. Koch and Friedlander describe the change in surface area as being driven by the difference in between the Area of the particles and some theoretical minimum Area,

$$\frac{dA(t)}{dt} = -\kappa(A(t) - A_f) \tag{1}$$

where A is the surface area of particles, κ is the characteristic sintering rate, and A_f is the final surface area of particles. In a 2-D model, A is perimeter of particles.

For the purposes of this example, we shall limit ourselves to the sintering of two particles. Moreover, the particles are of a simple geometry, namely squares. Thus, there are four variables that completely describe the state of the two particles—the size of each square and the position of the center of mass of each square. (Here we place the squares on the x-axis and neglect rotation, so that the center-of-mass is a one-dimensional variable.)

Since we have four unknowns, we require four equations. The sintering rate given in equation (1) is the first of these equations. The second equation is the conservation of the linear momentum, which for two particles centered on the x-axis, is simply,

$$m_1 \frac{dx_1}{dt} + m_2 \frac{dx_2}{dt} = 0 \tag{2}$$

where m_1 and m_2 are mass of two particles respectively. (Two particles are assumed to have the same density in this example). x_1 and x_2 are positions of the center of mass of two particles. The third equation is the conservation of mass.

$$\frac{dM}{dt} = 0 \tag{3}$$

where M is the total volume of two particles.

If the two particles are the same size then there are only three unknowns, the particle side length (for squares) or radius (for circles), and the positions of the center of mass for the two particles. The dynamic behavior of these three variables is completely specified by the three equations give above. When the particles are not the same size, we have a fourth variable, the size of the second particle, which requires a fourth equation. We have (somewhat arbitrarily) made the assumption that the rate of change of the volume normalized by the particle volume should be the same for each particle.

$$\frac{1}{V_1} \cdot \frac{dV_1}{dt} = \frac{1}{V_2} \cdot \frac{dV_2}{dt} \tag{4}$$

where V_1 and V_2 are volume of each particle at any given time. In a 2-D model, V_1 and V_2 are areas of two particles. This assumption is necessary to fully define the model. Certainly other assumptions on the relative rate of sintering in the two systems are also possible.

III. Derivation of Tractable Forms of the Equations:

We now have four equations and four unknowns, but the equations are written in terms of time derivatives for total area, *A*, total mass, *M*, volumes of each particle, V_1 and V_2, and the positions of the center of mass, x_1 and x_2. However, our independent unknowns are the size of each particle, l_1 and l_2, and the positions of the center of mass of each particle, x_1 and x_2. Therefore some reformulation of the ODEs is in order.

For the discussion that follows below, we consider two-dimensional squares. Where we use the term area, we mean perimeter and when we use the term volume, we mean area. In order to perform the reformulation of the equation, we need to invoke geometric constraints that take the form of nonlinear algebraic equations. The volume of each square is given by

$$V_1 = l_1^2 \qquad \text{and} \qquad V_2 = l_2^2 \tag{5}$$

The geometric definitions can be rendered into relationships between time derivatives through differentiation.

$$\frac{dV_1}{dt} = 2l_1 \frac{dl_1}{dt} \qquad \text{and} \qquad \frac{dV_2}{dt} = 2l_2 \frac{dl_2}{dt} \tag{6}$$

The two squares are initially placed adjacent to each other. During sintering, it is assumed that the two squares maintain their shape but now they overlap. Therefore the total volume of the system is

$$V = V_1 + V_2 - V_{over} \tag{7}$$

where the overlap volume, V_{over}, is a function of all four independent variables and can be determined by the examination of the geometry in the system, (See Figure 1.)

$$V_{over} = l_{over} \cdot l_{\min} = \left(\frac{l_1}{2} + \frac{l_2}{2} - \sqrt{(x_2 - x_1)^2} \right) \cdot l_{\min} \tag{8}$$

where l_{over} is the length of overlap and $l_{\min}$ is the smaller particle size, $l_{\min} = \min(l_1, l_2)$. The total mass is simply

$$M = \rho V \tag{9}$$

Therefore the conservation of mass is equivalent to the conservation of volume in this system,

$$\frac{dM}{dt} = \rho \frac{dV}{dt} = 0 = \frac{dV_1}{dt} + \frac{dV_2}{dt} - \tag{10}$$

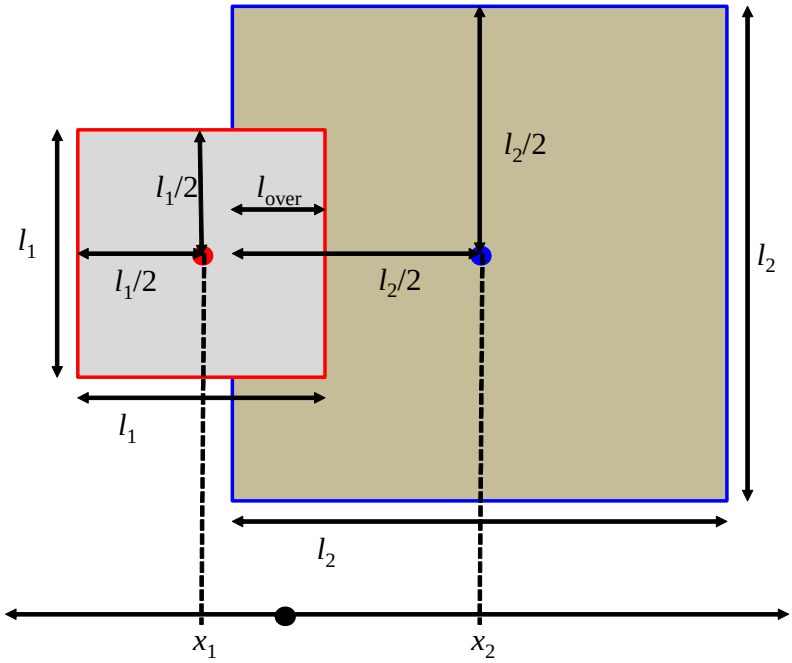

Figure 1. Geometry of the overlapping squares from which the distance of overlap can be determined from particle sizes and center of mass positions.

where the derivative of the overlap volume is obtained through the product rule,

$$\frac{dV_{over}}{dt} = l_{\min} \frac{dl_{over}}{dt} + l_{over} \frac{dl_{\min}}{dt} \tag{11}$$

We also need the area of the two particles, which can be obtained through the geometric analysis of the problem,

$$A = 4l_{\max} + 2\left(l_{\min} - l_{over}\right) \tag{12}$$

The time derivative of this area change is

$$\frac{dA}{dt} = 4\frac{dl_{\max}}{dt} + 2\left(\frac{dl_{\min}}{dt} - \frac{dl_{over}}{dt}\right) \tag{13}$$

The expression for the change in total area, equation (13), can be substituted into equation (1). The expression for the change in volumes, equation (6) and (11), can be substituted into equation (10) to yield

$$4\frac{dl_{\max}}{dt} + 2\left(\frac{dl_{\min}}{dt} - \frac{dl_{over}}{dt}\right) = -\kappa(A(t) - A_f) \tag{14}$$

$$2l_1\frac{dl_1}{dt} + 2l_2\frac{dl_2}{dt} = \frac{dl_{over}}{dt}\cdot l_{\min} + l_{over}\cdot\frac{dl_{\min}}{dt} \tag{15}$$

At this point, for simplicity we assume that $l_2 > l_1$ so that $l_{\min} = l_1$ and $l_{\max} = l_2$. We also evaluate the derivative of the overlap length,

$$\frac{dl_{over}}{dt} = \frac{1}{2}\frac{dl_1}{dt} + \frac{1}{2}\frac{dl_2}{dt} - \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right) \tag{16}$$

Substitution of equation (16) into equations (14) and (15) yield

$$4\frac{dl_2}{dt} + 2\left(\frac{dl_1}{dt} - \frac{1}{2}\frac{dl_1}{dt} - \frac{1}{2}\frac{dl_2}{dt} + \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right)\right) = -\kappa(A(t) - A_f) \tag{17}$$

$$2l_1\frac{dl_1}{dt} + 2l_2\frac{dl_2}{dt} = \left[\frac{1}{2}\frac{dl_1}{dt} + \frac{1}{2}\frac{dl_2}{dt} - \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right)\right]l_1 + l_{over}\frac{dl_1}{dt} \tag{18}$$

We can eliminate $\frac{dl_2}{dt}$ and $\frac{dx_2}{dt}$ as follows. Equation (4), providing sintering rate as a function of size is given by,

$$\frac{dl_2}{dt} = \frac{V_2}{V_1} \cdot \frac{l_1}{l_2}\frac{dl_1}{dt} = \frac{l_2}{l_1}\frac{dl_1}{dt} \tag{19}$$

The conservation of momentum is

$$\frac{dx_2}{dt} = -\frac{m_1}{m_2}\frac{dx_1}{dt} \tag{20}$$

So that equations (17) and (18) become

$$4\frac{l_2}{l_1}\frac{dl_1}{dt} + 2\left(\frac{dl_1}{dt} - \frac{1}{2}\frac{dl_1}{dt} - \frac{1}{2}\frac{l_2}{l_1}\frac{dl_1}{dt} + \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(-\frac{m_1}{m_2}\frac{dx_1}{dt} - \frac{dx_1}{dt}\right)\right) = -\kappa(A(t) - A_f) \tag{21}$$

$$2l_1\frac{dl_1}{dt} + 2l_2\frac{l_2}{l_1}\frac{dl_1}{dt} = \left[\frac{1}{2}\frac{dl_1}{dt} + \frac{1}{2}\frac{l_2}{l_1}\frac{dl_1}{dt} - \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(-\frac{m_1}{m_2}\frac{dx_1}{dt} - \frac{dx_1}{dt}\right)\right]l_1 + l_{over}\frac{dl_1}{dt} \tag{22}$$

Simplification yields,

$$\left(3\frac{l_2}{l_1} + 1\right)\frac{dl_1}{dt} - 2\frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(1 + \frac{m_1}{m_2}\right)\frac{dx_1}{dt} = -\kappa(A(t) - A_f) \tag{23}$$

$$\left(\frac{3}{2}l_1 - \frac{1}{2}l_2 + 2\frac{l_2^2}{l_1} - l_{over}\right)\frac{dl_1}{dt} = l_1\frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(1 + \frac{m_1}{m_2}\right)\frac{dx_1}{dt} \tag{24}$$

We can use equation (24) to solve for $\frac{dl_1}{dt}$

$$\frac{dl_1}{dt} = \frac{l_1\frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2}}\left(1 + \frac{m_1}{m_2}\right)}{\left(\frac{3}{2}l_1 - \frac{1}{2}l_2 + 2\frac{l_2^2}{l_1} - l_{over}\right)}\frac{dx_1}{dt} \tag{25}$$

We can substitute equation (25) into equation (23) to obtain our final equation,

$$\left[\left(3\frac{l_2}{l_1}+1\right)\frac{l_1\dfrac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2}}\left(1+\dfrac{m_1}{m_2}\right)}{\left(\dfrac{3}{2}l_1-\dfrac{1}{2}l_2+2\dfrac{l_2^2}{l_1}-l_{over}\right)}-2\frac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2}}\left(1+\frac{m_1}{m_2}\right)\right]\frac{dx_1}{dt}=-\kappa(A(t)-A_f) \quad (26)$$

At this point, the four equations can be expressed as ODEs in terms of the four unknowns,

$$\left[\left(3\frac{l_2}{l_1}+1\right)\frac{l_1\dfrac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2}}\left(1+\dfrac{m_1}{m_2}\right)}{\left(\dfrac{3}{2}l_1-\dfrac{1}{2}l_2+2\dfrac{l_2^2}{l_1}-l_{over}\right)}-2\frac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2}}\left(1+\frac{m_1}{m_2}\right)\right]\frac{dx_1}{dt}=-\kappa(A(t)-A_f) \quad (26)$$

$$\frac{dl_1}{dt}=\frac{l_1\dfrac{(x_2-x_1)}{\sqrt{(x_2-x_1)^2}}\left(1+\dfrac{m_1}{m_2}\right)}{\left(\dfrac{3}{2}l_1-\dfrac{1}{2}l_2+2\dfrac{l_2^2}{l_1}-l_{over}\right)}\frac{dx_1}{dt} \quad (25)$$

$$\frac{dl_2}{dt}=\frac{V_2}{V_1}\cdot\frac{l_1}{l_2}\frac{dl_1}{dt}=\frac{l_2}{l_1}\frac{dl_1}{dt} \quad (19)$$

$$\frac{dx_2}{dt}=-\frac{m_1}{m_2}\frac{dx_1}{dt} \quad (20)$$

IV. Parameter Specifications

We can numerically evaluate this problem given the initial conditions. The squares initially have sides of length 1 and 2.

$$l_1(t=0)=1 \qquad \text{and} \qquad l_2(t=0)=1 \quad (27)$$

The center of mass of the system is at zero and the particles are just touching, which leads to

$$x_1(t=0)=-1.2 \qquad \text{and} \qquad x_2(t=0)=0.3 \quad (28)$$

We use the following parameters, the density is one, $\rho=1$, and the sintering rate is $\kappa=5$.

V. Numerical Solution

We solve the method from time = 0 to 0.5 using the classical fourth-order Runge-Kutta method with 100 time intervals. The plots from this solution are reproduced below

As can be observed in Figure 2, both particles grow in size in time. This is possible because the center of mass of the two particles are drawing closer to each other, increasing the volume of overlap, and allowing both particles to grow, while maintaining a constant total particle volume.

A plot of the system center of mass (not shown) shows that the center of mass remained at zero, confirming that momentum was conserved during the sintering process.

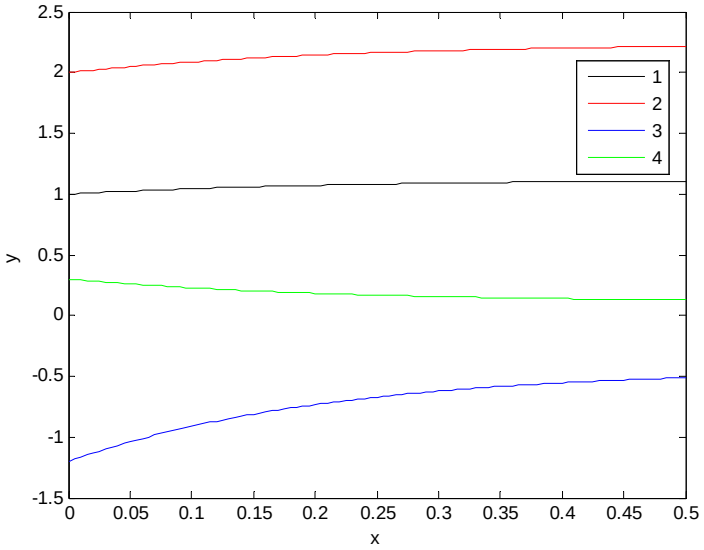

Figure 2. Plots of the four independent variables, the size of particle 1, l_1, (black) the size of particle 2, l_2, (red), the center of mass position of particle 1, x_1 (blue) and the center of mass position of particle 2, x_2, (green) as a function of time.

In Figure 3, a plot of the volume of the system is shown. Since the density is constant, the conservation of mass equates to a conservation of volume. We observe that the total volume is conserved. The volume of overlap between the two particles grows monotonically in time.

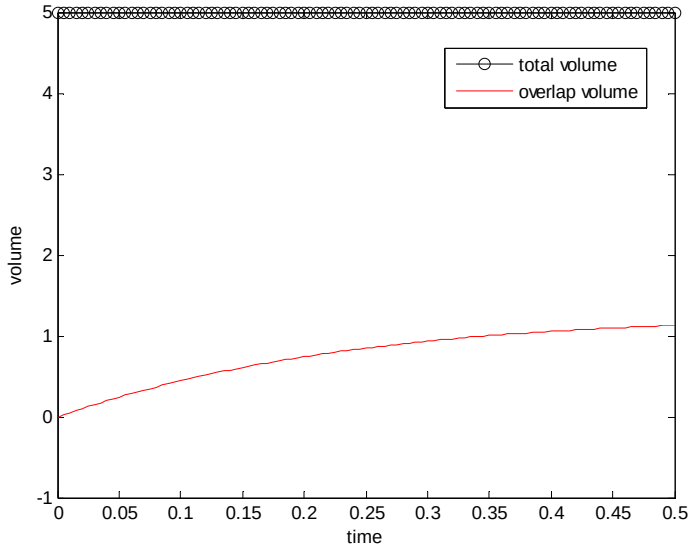

Figure 3. Volume as a function of time.

In Figure 4(a), we provide an initial configuration for the two particles. Their surfaces are just touching. In Figure 4(b) and 4(c) , the configuration is shown at times of 0.25 and 0.5 respectively. In these snapshots, one can observe both the growth in the size of each square as well as the coming together of the two center of masses.

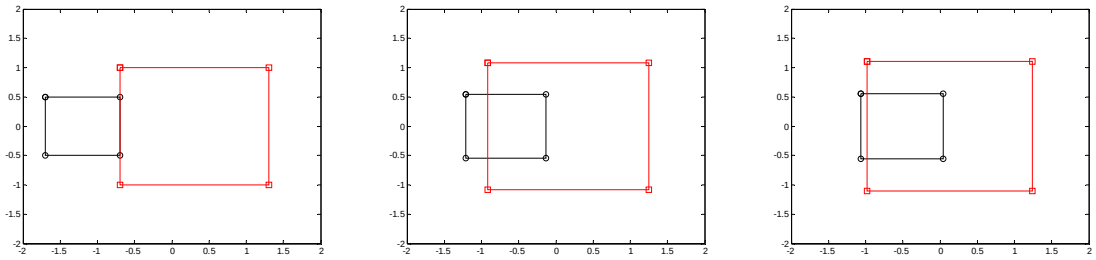

Figure 4. Snapshots of the particle configurations at (from left to right) t = 0, 0.25 and 0.5

In Figure 5., we provide a series of overlaid configurations in which the blue curve depicts the outline of the combined object and the dotted red and black lines represent the surfaces of the large and small particles respectively that appear within the volume of the other particle. Again it is clear that both particles are growing in size and that the center of mass of the particles is moving closer together.

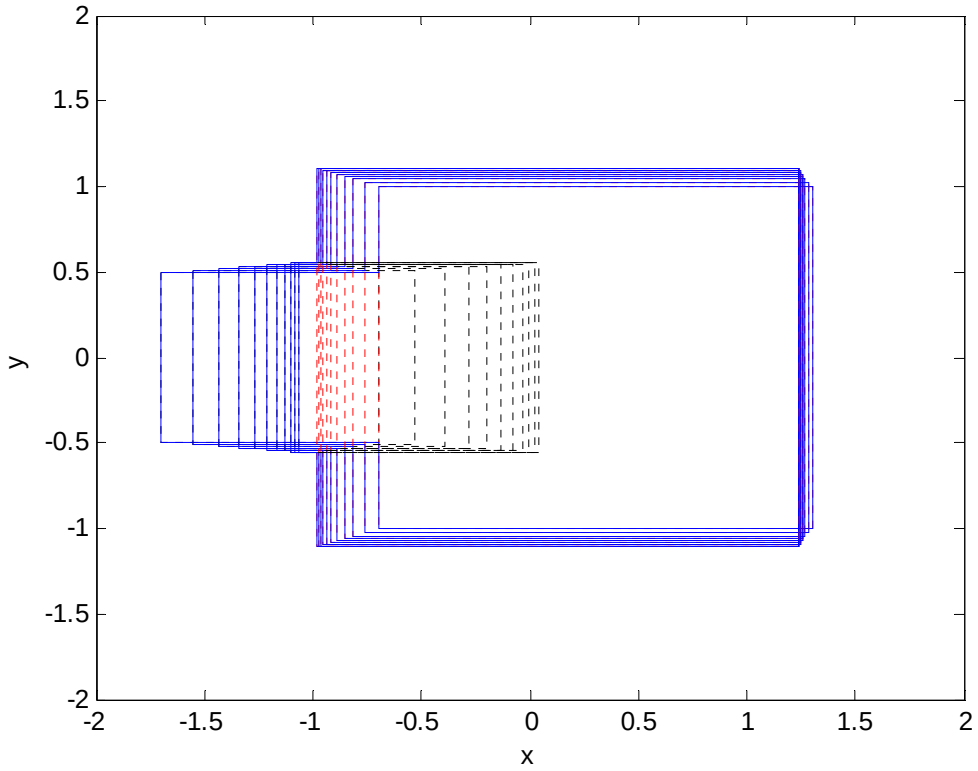

Figure 5. A series of overlaid configurations in which the blue curve depicts the outline of the combined object and the dotted red and black lines represent the surfaces of the large and small particles respectively that appear within the volume of the other particle.

The codes used to implement this numerical solution are provided in Appendix I.

Acknowledgements

These notes are based on a project performed by Zhao Wang, an undergraduate in the Department of Chemical Engineering, in 2006-7 at the University of Tennessee, under the advising of David Keffer.

Appendix I. Codes

This handout used a main program, driver.m, and an implementation of the classical fourth order Runge-Kutta method, rk4n.m.

I.A. driver.m

```
function driver
clear all;
close all
%
%  This function describes the sintering of two squares
%
%  This code was written by David Keffer and Zhao Wang in 2006.
%  University of Tennessee
global mass k Af
%
%  define initial sizes of squares
%  (This code assumes that particle 2 is bigger than particle 1.)
%
sideo(1) = 1;
sideo(2) = 2;
%
%  define volume and final surface areas
%
volo(1:2) = sideo(1:2).*sideo(1:2);
volf = sum(volo(1:2));
sidef = sqrt(volf);
Af = 4*sidef;
%
%  define density and mass
%
density = 1;
k = 5; % inverse time
mass(1:2) = density*volo(1:2);
%
%  determine initial location of center of mass of each square on x-axis
%
initial_sep=(sideo(1)+sideo(2))/2;
q(1) = -mass(2)/(mass(1)+mass(2))*initial_sep;
q(2) = mass(1)/(mass(1)+mass(2))*initial_sep;
%
%  solve ODEs describing dynamics of sintering
%
%  The four variables are
%  (1) size of first square
%  (2) size of second square
%  (3) position of cetner of mass of first square
%  (4) position of center of mass of second square
%
%  number of time steps
n = 100;
%  use Runge-Kutta Classical Fourth Order Method
xo = 0;
xf = 0.5;
yo = [sideo(1), sideo(2), q(1),q(2)];
[t,y]=rk4n(n,xo,xf,yo);
%
%  check conservation of momentum
%
for i = 1:1:n+1
    com(i) = mass(1)*y(i,3) + mass(2)*y(i,4);
```

```
end
figure(2);
plot(t,com,'r-');
xlabel('time');
ylabel('center of mass position');
%
%  check conservation of mass
%
for i = 1:1:n+1
    if (y(i,1) <= y(i,2) )
        miny = y(i,1);
    else
        miny = y(i,2);
    end
    dis = sqrt( (y(i,3)-y(i,4))^2 );
    fac = y(i,3)-y(i,4)/dis;
    v_overlap(i) = [y(i,1)/2 + y(i,2)/2 - dis]*min(y(i,1),y(i,2));
    v(i) = y(i,1)^2 + y(i,2)^2 - v_overlap(i);
end
figure(3);
plot(t,v,'k-o');
hold on;
plot(t,v_overlap,'r-');
hold off;
xlabel('time');
ylabel('volume');
legend('total volume','overlap volume');
%
%  make a movie with 2 rectangles
%
for i = 1:1:n+1
    ym1(1,i) = y(i,1)*0.5;
    ym1(2,i) = ym1(1,i);
    ym1(3,i) = -ym1(1,i);
    ym1(4,i) = ym1(3,i);
    ym1(5,i) = ym1(1,i);
    xm1(1,i) = y(i,3) - y(i,1)*0.5;
    xm1(2,i) = y(i,3) + y(i,1)*0.5;
    xm1(3,i) = xm1(2,i);
    xm1(4,i) = xm1(1,i);
    xm1(5,i) = xm1(1,i);

    ym2(1,i) = y(i,2)*0.5;
    ym2(2,i) = ym2(1,i);
    ym2(3,i) = -ym2(1,i);
    ym2(4,i) = ym2(3,i);
    ym2(5,i) = ym2(1,i);
    xm2(1,i) = y(i,4) - y(i,2)*0.5;
    xm2(2,i) = y(i,4) + y(i,2)*0.5;
    xm2(3,i) = xm2(2,i);
    xm2(4,i) = xm2(1,i);
    xm2(5,i) = xm2(1,i);
end
figure(4);
for i = 1:1:n+1
%for i = 1:1:1
    plot(xm1(:,i),ym1(:,i),'k-o');
    hold on;
    plot(xm2(:,i),ym2(:,i),'r-s');
    hold off;
    axis([-2 2 -2 2]);
   pause(0.1);
end
%
%  make a movie with total outline
%
for i = 1:1:n+1
```

```
    ym1(1,i) = y(i,1)*0.5;
    ym1(2,i) = ym1(1,i);
    ym1(3,i) = -ym1(1,i);
    ym1(4,i) = ym1(3,i);
    ym1(5,i) = ym1(1,i);
    xm1(1,i) = y(i,3) - y(i,1)*0.5;
    xm1(2,i) = y(i,3) + y(i,1)*0.5;
    xm1(3,i) = xm1(2,i);
    xm1(4,i) = xm1(1,i);
    xm1(5,i) = xm1(1,i);

    ym2(1,i) = y(i,2)*0.5;
    ym2(2,i) = ym2(1,i);
    ym2(3,i) = -ym2(1,i);
    ym2(4,i) = ym2(3,i);
    ym2(5,i) = ym2(1,i);
    xm2(1,i) = y(i,4) - y(i,2)*0.5;
    xm2(2,i) = y(i,4) + y(i,2)*0.5;
    xm2(3,i) = xm2(2,i);
    xm2(4,i) = xm2(1,i);
    xm2(5,i) = xm2(1,i);

    xmt(1,i) = xm1(1,i);
    ymt(1,i) = ym1(1,i);
    xmt(2,i) = xm2(1,i);
    ymt(2,i) = ymt(1,i);
    xmt(3,i) = xmt(2,i);
    ymt(3,i) = ym2(1,i);
    xmt(4,i) = xm2(2,i);
    ymt(4,i) = ymt(3,i);
    xmt(5,i) = xmt(4,i);
    ymt(5,i) = ym2(3,i);
    xmt(6,i) = xm2(4,i);
    ymt(6,i) = ymt(5,i);
    xmt(7,i) = xmt(6,i);
    ymt(7,i) = ym1(3,i);
    xmt(8,i) = xm1(1,i);
    ymt(8,i) = ymt(7,i);
    xmt(9,i) = xm1(1,i);
    ymt(9,i) = ym1(1,i);
end
figure(5);
nskip=10;
for i = 1:nskip:n+1
    plot(xm1(:,i),ym1(:,i),'k:');
    hold on;
    plot(xm2(:,i),ym2(:,i),'r:');
    hold on;
    plot(xmt(:,i),ymt(:,i),'b-');
    hold on;
    axis([-2 2 -2 2]);
   pause(0.1);
end
hold off;
xlabel('x');
ylabel('y');
%
%  make a snapshot with several total outline
%
figure(6);
nskip=20;
for i = 1:nskip:n+1
    plot(xmt(:,i),ymt(:,i),'b-');
    hold on;
    axis([-2 2 -2 2]);
end
hold off;
```

```
xlabel('x');
ylabel('y');
```

I.B. rk4n.m

```
function [x,y]=rk4n(n,xo,xf,yo);
%
%  [x,y]=rk4n(n,xo,xf,yo);
%
% This script will solve n first-order ODEs using
% the classical fourth-order Runge-Kutta method.
%
% ODE:  dy_i/dx = f_i(x,y_j)
%
% IC:   y_i(xo) = yo_i
%
% inputs:
% n = number of intervals, scalar
% x0 = value of x at initial condition, scalar
% xf = final value of x, scalar
% y0 = value of y at initial condition, vector of length m (m = # of ODEs)
% f_i(x) entered in the function "funkeval" at the bottom of this file
%
% outputs:
%
% x = vector of x values, nx1 vector
% y = matrix of y values, nxm vector
% a graph of y vs x
% a file, 'rk4n_out.txt' containing tabulated data of x and y
%
% example usage:
%
%  [x,y]=rk4n(10,0,2,[1,0,2]);
%
% Author:  David Keffer
% original date: October 23, 1998
% last modified date: April 2, 2013
%

%
% define discretization and x vector
%
dx = (xf-xo)/n;
x = zeros(n+1,1);
for i = 1:1:n+1
   x(i) = xo + (i-1)*dx;
end

%
% determine number of ODEs
%
m=max(size(yo));

%
% perform Runge-Kutta steps
%
y = zeros(n+1,m);
y(1,1:m) = yo(1:m);
dydx = zeros(1,m);
ytemp = zeros(1,m);
k1 = zeros(1,m);
k2 = zeros(1,m);
k3 = zeros(1,m);
k4 = zeros(1,m);
for i =  1:1:n
```

```
   x1 = x(i);
   ytemp(1:m) = y(i,1:m);
   k1(1:m) = funkeval(x1,ytemp);
   x2 = x(i) + 0.5*dx;
   ytemp(1:m) = y(i,1:m) + 0.5*dx*k1(1:m);
   k2(1:m) = funkeval(x2,ytemp);
   x3 = x(i) + 0.5*dx;
   ytemp(1:m) = y(i,1:m) + 0.5*dx*k2(1:m);
   k3(1:m) = funkeval(x3,ytemp);
   x4 = x(i) + dx;
   ytemp(1:m) = y(i,1:m) + dx*k3(1:m);
   k4(1:m) = funkeval(x4,ytemp);
   dydx(1:m) = 1.0/6.0*(k1(1:m) + 2.0*k2(1:m) + 2.0*k3(1:m) + k4(1:m));
   y(i+1,1:m) = y(i,1:m) + dx*dydx(1:m);
end

%
%  plot
%
close all;
iplot = 1;
if (iplot == 1)
   for i = 1:1:m
      color_index = get_plot_color(i);
      plot (x(:),y(:,i),color_index);
      hold on;
   end
   hold off;
   xlabel( 'x' );
   ylabel ( 'y' );
   legend (int2str([1:m]'));
end

%
%  write result to file 'rk4n_out.txt'
%
fid = fopen('rk4n_out.txt','w');
fprintf(fid,'x  y(1) ... y(m) \n');
for i = 1:1:n+1
   fprintf(fid,'%23.15e ', x(i));
   for j = 1:1:m
      fprintf(fid,'%23.15e ', y(i,j));
   end
   fprintf(fid,' \n');
end
fclose(fid);

%
% enter ODE in this function
%
function dydt = funkeval(t,y);
%
%  The sintering of two squares
%
global mass k Af
%
%  translate function inputs to sintering variables
%
side1 = y(1);
side2 = y(2);
x1 = y(3);
x2 = y(4);
%
%  determine volumes and areas
%
V1 = side1*side1;
V2 = side2*side2;
```

```
sidemin = min(side1,side2);
sidemax = max(side1,side2);
dis = x2 - x1;
dissr = sqrt((dis)^2);
sideover = side1/2 + side2/2 - dissr;
A = 4*sidemax + 2*(sidemin-sideover);
Vover = sideover*sidemin;
V = V1 + V2 - Vover;
%
%  thermodynamically driven sintering rate law
%
numerator1 = side1*dis/dissr*(1+mass(1)/mass(2));
denominator1 = 3/2*side1 - 1/2*side2 + 2*side2^2/side1 - sideover;
fac1 = 3*side2/side1 + 1;
term1 = fac1*numerator1/denominator1;
term2 = -2*dis/dissr*(1+mass(1)/mass(2));
denominator2 = term1 + term2;
numerator2 = -k*(A - Af);
dx1dt = numerator2/denominator2;
%
%  conservation of mass
%
dside1dt = numerator1/denominator1*dx1dt;
%
%  relative sintering ratios
%
dside2dt = side2/side1*dside1dt;
%
%  conservation of momentum
%
dx2dt = -mass(1)/mass(2)*dx1dt;
%
%  copy the derivatives to the out-going vector
%
dydt(1) = dside1dt;
dydt(2) = dside2dt;
dydt(3) = dx1dt;
dydt(4) = dx2dt;

%
% this little function sets colors for curves in plot
%
function color_index = get_plot_color(i);
if (i == 1)
   color_index = 'k-';
elseif (i == 2)
   color_index = 'r-';
elseif (i == 3)
   color_index = 'b-';
elseif (i == 4)
   color_index = 'g-';
elseif (i == 5)
   color_index = 'm-';
elseif (i == 6)
   color_index = 'k:';
elseif (i == 7)
   color_index = 'r:';
elseif (i == 8)
   color_index = 'b:';
elseif (i == 9)
   color_index = 'g:';
elseif (i == 10)
   color_index = 'm:';
else
   color_index = 'k-';
end
```