Catam Additional Projects
Computational Projects Manual
(July 2019 Edition)

4/15/2020

- Collapse of a Spherical Cavitation Bubble
- Programs

1 Collapse of a Spherical Cavitation Bubble

We have

$$p(r,t) = 1 + \frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}t}(R^2\frac{\mathrm{d}R}{\mathrm{d}t}) - \frac{R^4}{2r^4}(\frac{\mathrm{d}R}{\mathrm{d}t})^2 \quad with \quad b.c. \quad p(R,t) = -\frac{2\lambda}{R}. \tag{$*$}$$

Hence

$$p(R,t) = 1 + \frac{1}{R}\frac{\mathrm{d}}{\mathrm{d}t}(R^2\frac{\mathrm{d}R}{\mathrm{d}t}) - \frac{1}{2}(\frac{\mathrm{d}R}{\mathrm{d}t})^2 \tag{$**$}$$

1.1 Question 1

Solve $(**)$ for R(t) by multiplying by $2R^2\dot{R}$ on both sides and rearrange,

$$(-4\lambda R - 2R^2)\dot{R} = \frac{\mathrm{d}}{\mathrm{d}t}(R^3\dot{R}^2) \tag{1}$$

The bubble starts rest and has unit initial radius, so $\dot{R}(0) = 0$, $R(0) = 1$. Insert the initial conditions into (1) to get

$$\dot{R}^2 = \frac{2}{3}(\frac{1}{R^3} - 1) + 2\lambda(\frac{1}{R^3} - \frac{1}{R}) \tag{2}$$

as required.

Note that $\dot{R}^2 \geq 0 \Rightarrow (1-R)(\frac{2}{3}(1+R+R^2) + 2\lambda(1+R)) \geq 0$ so $0 \leq R \leq 1$.

Plot $\mathrm{d}R/\mathrm{d}t$ against R and $\log(\mathrm{d}R/\mathrm{d}t)$ against log(R) for $\lambda = 0.0, 0.1, 1.0, 10.0$, and 100.0. Programs for plots are listed on page 12.

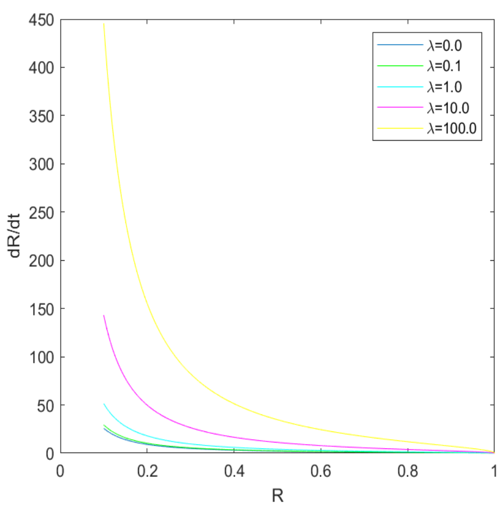

Figure 1: $\mathrm{d}R/\mathrm{d}t$ against R for $\lambda = 0.0, 0.1, 1.0, 10.0,$ and 100.0

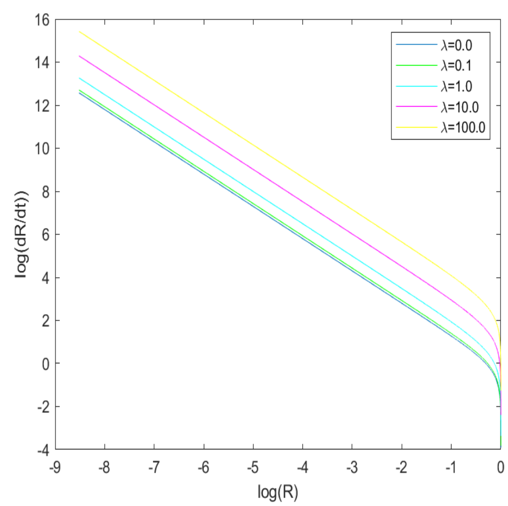

Figure 2: Scaled $\mathrm{d}R/\mathrm{d}t$ against R for $\lambda = 0.0, 0.1, 1.0, 10.0,$ and 100.0

From Fig.2, we can see that

- As λ rises, log(dR/dt) shifts upwards, i.e., R changes more rapidly: the bubble with higher surface tension shrinks more quickly.
- For individual curve, log(dR/dt) decreases almost linearly as log(R) increases from 0 except when R is close to 1, where the curve becomes concave downwards and $\mathrm{d}R/\mathrm{d}t \to 0$ rapidly as R $\to$ 1: the bubble starts from rest and shrinks at a faster and faster speed.

1.2 Question 2

From $(**)$ and (2) we get

$$\ddot{R} = \frac{\lambda}{R^2} - \frac{1}{R^4} - \frac{3\lambda}{R^4} \quad and \quad \dot{R}^2 = \frac{2}{3R^3}(1 - R^3 + 3\lambda(1 - R^2)) = \frac{2}{3R^3}\beta. \tag{3}$$

where $\beta = 1 - R^3 + 3\lambda(1 - R^2) > 0$ since $0 \leq R \leq 1, \lambda \geq 0$.

$$\begin{aligned} (*)(3) \Rightarrow p(r, R) &= 1 + \frac{1}{r}(\frac{3\lambda + 1}{3R^2} - \frac{4R}{3} - 3\lambda) - \frac{R}{3r^4}\beta \\ &= 1 + \frac{1}{r}(\frac{1}{3R^2}(1 - 4R^3) + \frac{\lambda}{R^2}(1 - 3R^2)) - \frac{R}{3r^4}\beta \\ &= 1 + \frac{4}{3rR^2}\alpha - \frac{R}{3r^4}\beta. \end{aligned} \tag{4}$$

where $\alpha = \frac{1}{4}(1 - 4R^3) + \frac{3\lambda}{4}(1 - 3R^2)$.

• Case $\alpha < 0$,

Since $\beta > 0 \Rightarrow -\frac{R}{3r^4}\beta < 0$, p is maximized when $\frac{4}{3rR^2}\alpha - \frac{R}{3r^4}\beta < 0$ is minimized. Since p is bounded above by 1, p is maximized at infinity, i.e., $p_{max} = 1$ when r $\to \infty$.

• Case $\alpha > 0$,

$$\frac{\mathrm{d}p}{\mathrm{d}r} = -\frac{4}{3r^2R^2}\alpha + \frac{4R}{3r^5}\beta = 0 \Rightarrow r = \sqrt[3]{\frac{\beta}{\alpha}}R \quad and \quad \left.\frac{\mathrm{d}^2p}{\mathrm{d}r^2}\right|_{r=\sqrt[3]{\frac{\beta}{\alpha}}R} = -\frac{4\alpha^2}{\beta R^5} < 0.$$

Hence

$$p_{max} = 1 + \frac{1}{R^3}\sqrt[3]{\frac{\alpha^4}{\beta}}.$$

In summary,

$$p_{max} = \begin{cases} 1 & \alpha < 0 \\ 1 + \frac{1}{R^3}\sqrt[3]{\frac{\alpha^4}{\beta}} & \alpha > 0 \end{cases} . \tag{5}$$

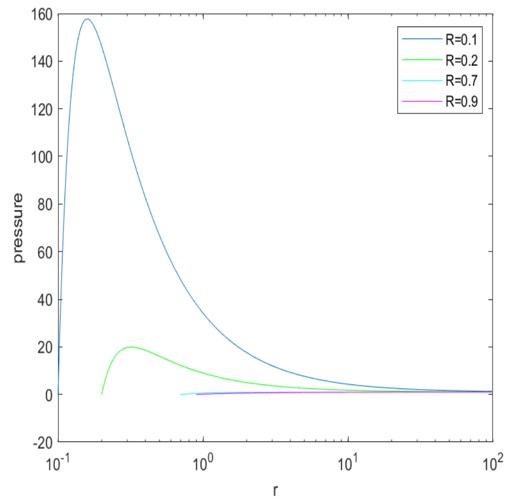

Figure 3: p(r,R) against r when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 0$

R	p_{max}	$r = \underset{r>R}{argmax}\, p$
0.1	157.70300	0.15890
0.2	19.90133	0.31976
0.7	1	∞
0.9	1	∞

Table 1: p_{max} and $r = \underset{r>R}{argmax}\, p$ when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 0$

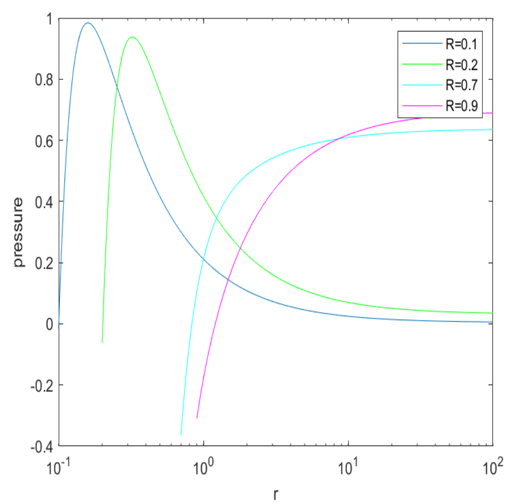

Figure 4: p(r,R) against r when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 0.2$ (normalised)

R	p_{max}	$r = \underset{r>R}{argmax}\, p$
0.1	248.73516	0.15894
0.2	29.99276	0.32275
0.7	1	∞
0.9	1	∞

Table 2: p_{max} and $r = \underset{r>R}{argmax}\, p$ when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 0.2$

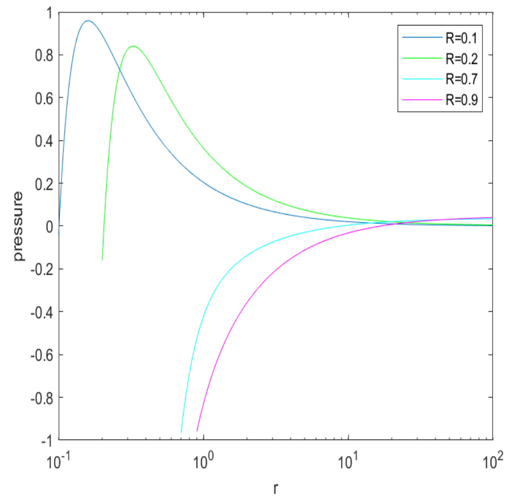

Figure 5: p(r,R) against r when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 9.0$ (normalised)

R	p_{max}	$r = \underset{r>R}{argmax}\, p$
0.1	4254.38723	0.15994
0.2	474.25995	0.32675
0.7	1	∞
0.9	1	∞

Table 3: p_{max} and $r = \underset{r>R}{argmax}\, p$ when $R = 0.1, 0.2, 0.7, 0.9$ for $\lambda = 9.0$

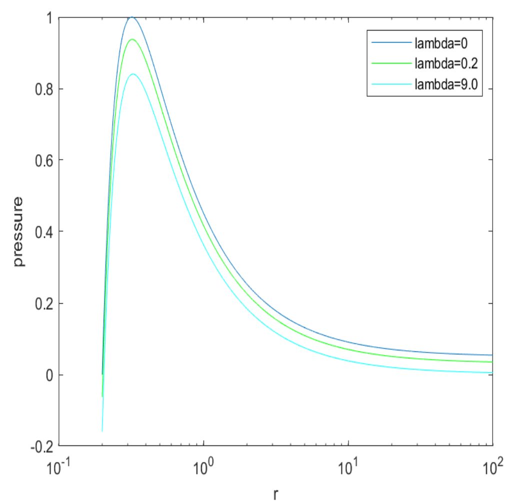

Figure 6: $\lambda = 0, 0.2, 9.0$ when $R = 0.2$ (normalised)

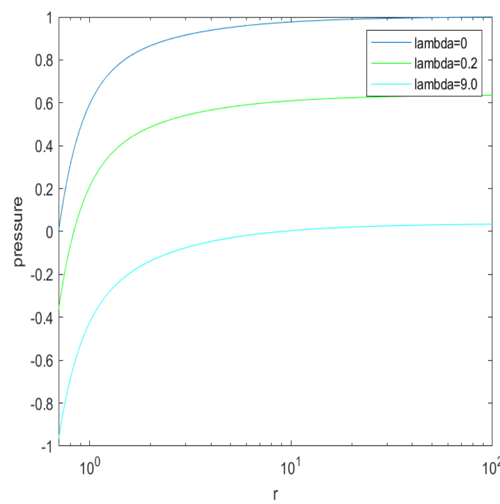

Figure 7: $\lambda = 0, 0.2, 9.0$ when $R = 0.7$ (normalised)

Some plots between different R (holding λ constant) (Fig.3-5), and different λ (holding R constant) (Fig.6-7) are shown above with corresponding p_{max} and $\underset{r>R}{argmax}\, p$ shown on the left (Tbl.1-3). Related Programs are listed on page 13, named **Q2plot**.

In the case $\alpha > 0$, pressure first increases and peaks at a point which goes lower as R rises, then drops to 1 at infinity with slowing speed. There is a shift towards right of the peak as R increases (i.e., $r = \underset{r>R}{argmax}\, p$ is larger). In the case $\alpha < 0$, r goes through larger distance before p peaks as R rises, the pressure also goes to 1 at finity.

Among different λ's, p_{max} rises as λ rises for $\alpha > 0$. This is consistent with (5), since α (hence p_{max}) is an increasing function of λ. There is also a slight right shift of $\underset{r>R}{argmax}\, p$ as λ inreases (while holding R constant).

From normalized plots for $\alpha > 0$ (e.g. R=0.2) and $\alpha < 0$ (e.g. R=0.7) (Fig.6-7), we can observe a similar shape of p(r) between different λ's for each case. The pressure develops in the same pattern as we go away from the bubble surface and remains flat at infinity.

1.3 Question 3

1.3.1 Numerical Solution for R(t): Programming Task

A program solving (2) is listed on page 15, named **Q3Eulerii(t0,h,p,tmax)**. t_0, t_{max} is the start-point and the end-point of the time range we choose, h is the step size, p stands for λ. Here we use $t_0 = 1 \times 10^{-6}, h = 1 \times 10^{-6}, t_{max} = 1$ with $\lambda = 0$.

To avoid the trivial solution R $\equiv$ 1, we will find a series solution for R for small t as the first step.

First notice that, when we substitute $x = R^{5/2}$ into (2), we get

$$\frac{4}{25}x^{-\frac{6}{5}}\dot{x}^2 = \frac{2}{3}(x^{-\frac{6}{5}} - 1) + 2\lambda(x^{-\frac{6}{5}} - x^{-\frac{2}{5}}) \quad \Rightarrow \dot{x}^2 = \frac{25}{6}(1 - x^{\frac{6}{5}}) + \frac{25}{2}\lambda(1 - x^{\frac{4}{5}}).$$

Hence

$$\dot{x} = -\frac{5}{2}(\frac{2}{3}(1 - x^{\frac{6}{5}}) + 2\lambda(1 - x^{\frac{4}{5}}))^{\frac{1}{2}} \tag{6}$$

$$\ddot{x} = (\frac{2}{3}(1 - x^{\frac{6}{5}}) + 2\lambda(1 - x^{\frac{4}{5}}))^{-\frac{1}{2}}(x^{\frac{1}{5}} + 2\lambda x^{-\frac{1}{5}}) \tag{7}$$

with corresponding i.c. $x(0) = R(0) = 1$.

Here we choose the negative root because physically, the pressure outside the bubble push its surface inwards. $p = -2\lambda/R < 0$ at bubble surface implies force pointing into the bubble from the fluid. From a mathematical point of view, (6) only makes sense if expressions under the square root is non-negative, which means x can only decrease from 1 where it starts.

It barely makes sense if we choose the positive root, except when, for example, heat is applied to the fluid.

R=0 singular in (2) (due to its appearance on the denominator) implies x=0 singular in (6). Higher-order methods, e.g. Runge-Kutta, where $f(t_n+h/2, y_n+hk_1/2)$ ($k_1 = f(t_n, y_n) = \left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{(t_n,y_n)}$ and etc.) and other second derivatives are involved, will make the term $x^{-\frac{1}{5}}$ in (7) invalid as x $\to 0$. $\ddot{x}$ and hence f may diverge as x $\to 0$. On the other hand, Euler involves only first derivatives instead, which results in better accuracy.

Now, to find the series solution, set $x = 1 - \epsilon$ for small t ($\epsilon > 0$ since $\dot{x}(0) < 0$) and Taylor expand (6)

$$\begin{aligned}
-\dot{\epsilon} &= -\frac{5}{2}(\frac{2}{3}(1-(1-\frac{6}{5}\epsilon+\frac{\frac{6}{5}\times\frac{1}{5}}{2!}\epsilon^2+\mathcal{O}(\epsilon^3)))+2\lambda(1-(1-\frac{4}{5}\epsilon+\frac{\frac{4}{5}\times(-\frac{1}{5})}{2!}\epsilon^2+\mathcal{O}(\epsilon^3))))^{\frac{1}{2}} \\
&= -\frac{5}{2}((1+2\lambda)(\frac{4}{5}\epsilon+\frac{2}{25}\epsilon^2)+\mathcal{O}(\epsilon^3))^{\frac{1}{2}} \\
&= -\sqrt{5(1+2\lambda)\epsilon}+\mathcal{O}(\epsilon^{\frac{3}{2}}).
\end{aligned}$$

Separate variables and insert i.c. $\epsilon(0) = 0$ to get

$$\epsilon(t) \approx \frac{5(1+2\lambda)}{4}t^2$$

Hence the solution to (6) is

$$x(t) = 1 - \frac{5(1+2\lambda)}{4}t^2 \tag{8}$$

for small t.

At this stage we can solve (6) numerically using Euler method and (8) for the first step, but let′s try solving (6) analytically for the case $\lambda = 0$ for further confirmation.

Use the substitution $x = sin^{\frac{5}{3}}\theta$. Since $x_0 = 1$, $x_c = 0$ (bubble collapse when $x = R = 0$), let $\theta_0 = \pi/2$, $\theta_c = \pi$. We get

$$\begin{aligned}
\dot{x} = \frac{5}{3}sin^{\frac{2}{3}}\theta cos\theta\dot{\theta} &= -\frac{5}{2}(\frac{2}{3}(1-sin^2\theta)+2\lambda(1-sin^{\frac{4}{3}}\theta))^{\frac{1}{2}} \\
\Rightarrow \dot{\theta} &= -\frac{3}{2}\frac{1}{sin^{\frac{2}{3}}cos\theta}(\frac{2}{3}cos^2\theta+2\lambda(1-sin^{\frac{4}{3}}\theta))^{\frac{1}{2}} \\
\Rightarrow \dot{\theta} &= \sqrt{\frac{3}{2}}\frac{1}{sin^{\frac{2}{3}}\theta} \qquad (\lambda = 0)
\end{aligned} \tag{9}$$

by noticing $\cos\theta < 0$ for $\theta \in (\pi/2, \pi)$. To integrate (9), run the program listed on page 16, named **solvetc**, for numerical integration. We might as well apply gamma function and consider

$$\begin{aligned}\Gamma(m)\Gamma(n) &= \int_0^\infty \int_0^\infty x^{m-1}y^{n-1}e^{-(x+y)}\mathrm{d}x\mathrm{d}y\\ x=p^2, y=q^2 &\Rightarrow 4\int_0^\infty \int_0^\infty p^{2m-1}q^{2n-1}e^{-(p^2+q^2)}\mathrm{d}p\mathrm{d}q\\ p=r\cos\theta, q=r\sin\theta &\Rightarrow 4\int_0^{\pi/2} \int_0^\infty \cos^{2m-1}\theta\sin^{2n-1}\theta\mathrm{d}\theta r^{2m+2n-1}e^{-r^2}\mathrm{d}r\\ &= 2\int_0^{\pi/2} \cos^{2m-1}\theta\sin^{2n-1}\theta\mathrm{d}\theta\Gamma(m+n).\end{aligned}$$

Hence by separating variables,

$$(9) \Rightarrow \int_0^{t_c} 1\mathrm{d}t = \int_{\pi/2}^{\pi} \sqrt{\frac{2}{3}} \sin^{\frac{2}{3}}\theta\mathrm{d}\theta \quad \Rightarrow t_c = \sqrt{\frac{1}{6}}\frac{\Gamma(\frac{1}{3}+\frac{1}{2})\Gamma(\frac{1}{2})}{\Gamma(\frac{4}{3})} \approx 0.914681. \tag{10}$$

Run the program in §1.3.1 (i.e. Q3Eulerii$(1\times 10^{-6}, 1\times 10^{-6}, 0, 1)$), we get plots for R(t) and scaled $p_{max}(t)$ as well as the value of $t_c = 0.914689$.

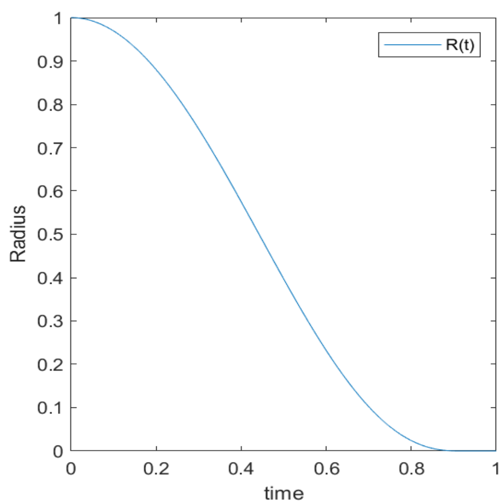

Figure 8: R(t) when $\lambda = 0$

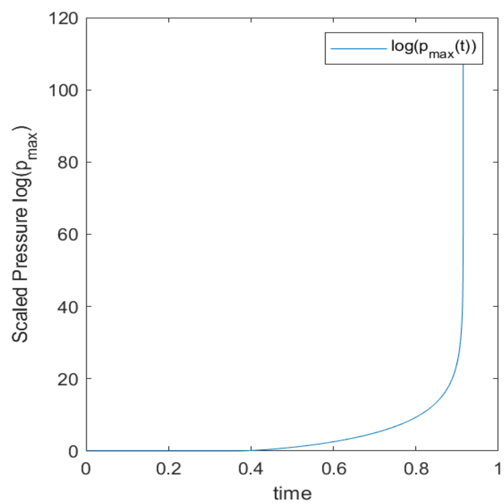

Figure 9: $log(p_{max}(t))$ when $\lambda = 0$

The percentage error of t_c as we calculated from the numerical calculations against the true value is approximately

$$\frac{0.914689 - 0.914681}{0.914681} \times 100\% = (8.7462 \times 10^{-4})\%.$$

The accuracy of the numerical calculations is satisfying. This rather small error may come from

- the approximation error when we use the series solution in (8) ignoring all $\mathcal{O}(\epsilon^{3/2})$ terms. E.g., if we start at $t = 1 \times 10^{-6}$, there is an error of $\mathcal{O}(10^{-18})$.
- the step we take in the program. Smaller h will result in smaller error.

1.4 Question 4

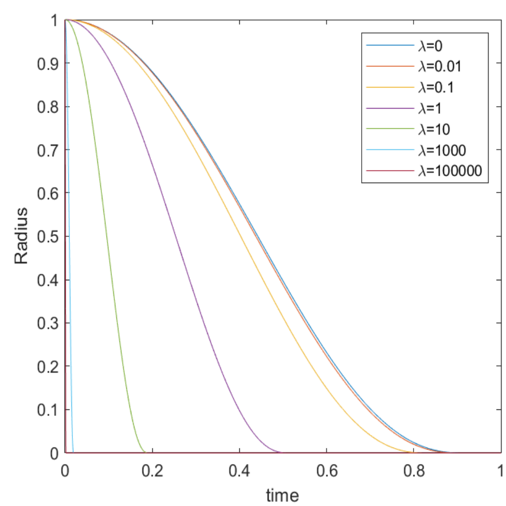

Figure 10: R(t) with different λ's

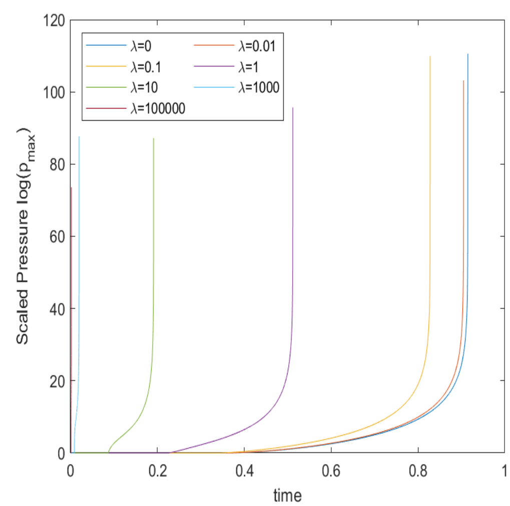

Figure 11: $log(p_{max}(t))$ with different λ's

λ	0	0.01	0.1	1	10	1000	100000
t_c	0.914689	0.904763	0.828006	0.511718	0.191102	0.019545	0.001959

Table 4: A list of λ's and corresponding t_c's from numerical computation

Programs for plotting R(t) and $log(p_{max}(t))$ are listed on page 17-19, named **Q4**, **Q4EulerR(t0,h,p,tmax)**, **Q4EulerP(t0,h,p,tmax)** and **Q4plot**.

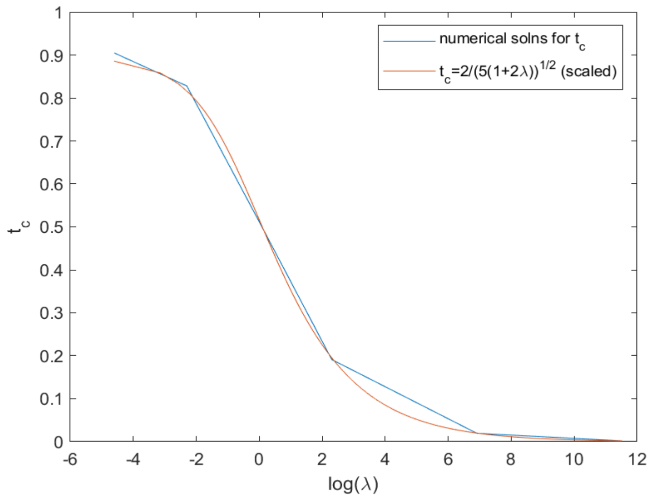

Figure 12: plot of t_c against $log(\lambda)$ from Tbl.4 compared with function (11)

Plot t_c against λ (scaled) from Tbl.4. Plots in blue shown in Fig.12. In order to approximate the plots, recall from (8), for small t,

$$x(t_c) = 0 \quad \Leftrightarrow \quad t_c = \frac{2}{\sqrt{5(1+2\lambda)}}. \tag{11}$$

It can be seen from Fig.12 that (11) (in red) is rather a good fit for the relationship between λ and t_c obtained from numerical calculations.

This relationship also makes sense in a physical way. Higher surface tension results in higher pressure at the boundary (see also $(*)$), which causes faster shrink of the bubble.

From Fig.10, R(t) is shown to be monotone decreasing from concave to convex (fast in the middle and slowly at the beginning and the end of the evolution). As λ increases, it drops more rapidly and hits 0 with small t_c. The maximum pressure shown in Fig.11 remains 1 before it climbs up and goes to infinity at t_c, which it does earlier and earlier as λ increases.

For $\lambda \ll 1$, $t_c \to 0.914689$ and p_{max} spreads out through time (i.e., it goes to infinity more slowly). For $\lambda \gg 1$, $t_c \to 0$ and p_{max} compresses (i.e., it goes to infinity right away after start). This can be shown in both Fig.10-11 and Fig.12.

1.5 Question 5

In the real world, the model described by ($*$) is flawed. There are several factors that will contribute to a more complex physical model. These factors includes but are not limited to

- the viscosity of the fluids;
- the presence of vapour inside bubbles;
- the non-spherical nature of the bubble during the collapse.

For example, in a boat propellor, we may estimate the initial radius of bubbles generated to be around $1mm$ (due to high fluid pressure). Assuming that the bubbles are empty, Fig.13 shows that, in the case when $\lambda = 0$, for example, $\dot{R}$ approaches the speed of light ($3 \times 10^8 m/s$) as R goes to around $1.95 \times 10^{-6}m$. Hence the model is no longer valid when R is below $1.95 \times 10^{-6}m$ for $\lambda = 0$.

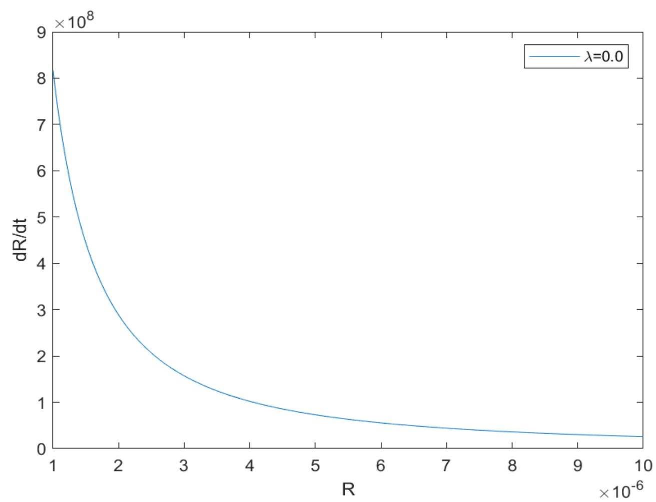

Figure 13: $\mathrm{d}R/\mathrm{d}t$ against R for $\lambda = 0.0, 0.1, 1.0, 10.0$, and 100.0

To modify the model in the case when bubbles contain small amounts of vapour (or other gas), extra terms and viscosity coefficient need to be added to ($*$). For example, we can apply Rayleigh-Plesset equation

$$\frac{p_b - p_\infty}{\rho} = R\frac{\mathrm{d}^2 R}{\mathrm{d}t^2} + \frac{3}{2}(\frac{\mathrm{d}R}{\mathrm{d}t})^2 + \frac{4\,v}{R}\frac{\mathrm{d}R}{\mathrm{d}t} + \frac{2\lambda}{\rho R} \tag{12}$$

where p_b is the pressure inside bubbles due to the gas content, p_∞ is the pressure at infinity, ρ is the fluid density outside and v is the viscosity of the fluid outside.

2 Programs

Note: Some programs listed on this pdf have $'return'$ added after excessively long texts for clarity, which needs to be removed before tested.

2.1 Question 1

2.1.1 Q1plot: plots of $\mathrm{d}R/\mathrm{d}t$ against R

```
R=linspace(0,1,5000);
DR1=(2/3*(1./R.^3-1)).^0.5;
DR2=(2/3*(1./R.^3-1)+0.2*(1./R.^3-1./R)).^0.5;
DR3=(2/3*(1./R.^3-1)+2*(1./R.^3-1./R)).^0.5;
DR4=(2/3*(1./R.^3-1)+20*(1./R.^3-1./R)).^0.5;
DR5=(2/3*(1./R.^3-1)+200*(1./R.^3-1./R)).^0.5;
plot(R,DR1,R,DR2,'g',R,DR3,'c',R,DR4,'m',R,DR5,'y')
xlabel('R'), ylabel('dR/dt')
legend('\lambda=0.0' , '\lambda=0.1','\lambda=1.0' ,
'\lambda=10.0','\lambda=100.0')
```

2.1.2 Q1plot: plots of log($\mathrm{d}R/\mathrm{d}t$) against log(R)

```
R=linspace(0,1,5000);
logDR1=1/2*log(2/3*(1./R.^3-1));
logDR2=1/2*log(2/3*(1./R.^3-1)+0.2*(1./R.^3-1./R));
logDR3=1/2*log(2/3*(1./R.^3-1)+2*(1./R.^3-1./R));
logDR4=1/2*log(2/3*(1./R.^3-1)+20*(1./R.^3-1./R));
logDR5=1/2*log(2/3*(1./R.^3-1)+200*(1./R.^3-1./R));
plot(log(R),logDR1,log(R),logDR2,'g',log(R),logDR3,'c'
,log(R),logDR4,'m',log(R),logDR5,'y')
xlabel('log(R)'), ylabel('log(dR/dt))')
legend('\lambda=0.0' , '\lambda=0.1','\lambda=1.0' ,
'\lambda=10.0','\lambda=100.0')
```

2.2 Question 2

Q2plot

```
% lambda=0
r1=linspace(0.1,100,100000);
p1=1+4/3*1./r1*1/0.1^2*1/4*(1-4*0.1^3)-0.1./(3*r1.^4)*(1-0.1^3);
a1=max(p1),[argvalue1, argmax1] = max(p1)
r2=linspace(0.2,100,100000);
p2=1+4/3*1./r2*1/0.2^2*1/4*(1-4*0.2^3)-0.2./(3*r2.^4)*(1-0.2^3);
a2=max(p2),[argvalue2, argmax2] = max(p2)
r3=linspace(0.7,100,100000);
p3=1+4/3*1./r3*1/0.7^2*1/4*(1-4*0.7^3)-0.7./(3*r3.^4)*(1-0.7^3);
a3=max(p3),[argvalue3, argmax3] = max(p3)
r4=linspace(0.9,100,100000);
p4=1+4/3*1./r4*1/0.9^2*1/4*(1-4*0.9^3)-0.9./(3*r4.^4)*(1-0.9^3);
a4=max(p4),[argvalue4, argmax4] = max(p4)
plot(r1,p1,r2,p2,'g',r3,p3,'c',r4,p4,'m')
xlabel('r'), ylabel('pressure')
legend('R=0.1' , 'R=0.2','R=0.7' , 'R=0.9')
set(gca,'Xscale','log')

% lambda=0.2
r1=linspace(0.1,100,100000);
p1=1+4/3*1./r1*1/0.1^2*(1/4*(1-4*0.1^3)+3/4*0.2*(1-3*0.1^2))-0.1./
(3*r1.^4)*(1-0.1^3+3*0.2*(1-0.1^2));
a1=max(p1);b1=min(p1);[argvalue1, argmax1] = max(p1)
r2=linspace(0.2,100,100000);
p2=1+4/3*1./r2*1/0.2^2*(1/4*(1-4*0.2^3)+3/4*0.2*(1-3*0.2^2))-0.2./
(3*r2.^4)*(1-0.2^3+3*0.2*(1-0.2^2));
a2=max(p2);b2=min(p2);[argvalue2, argmax2] = max(p2)
r3=linspace(0.7,100,100000);
p3=1+4/3*1./r3*1/0.7^2*(1/4*(1-4*0.7^3)+3/4*0.2*(1-3*0.7^2))-0.7./
(3*r3.^4)*(1-0.7^3+3*0.2*(1-0.7^2));
a3=max(p3);b3=min(p3);[argvalue3, argmax3] = max(p3)
r4=linspace(0.9,100,100000);
p4=1+4/3*1./r4*1/0.9^2*(1/4*(1-4*0.9^3)+3/4*0.2*(1-3*0.9^2))-0.9./
(3*r4.^4)*(1-0.9^3+3*0.2*(1-0.9^2));
a4=max(p4);b4=min(p4);[argvalue4, argmax4] = max(p4)
```

```
plot(r1,p1/(a1-b1),r2,p2/(a2-b2),'g',r3,p3/(a3-b3),'c',r4,p4/(a4-b4),'m')
xlabel('r'), ylabel('pressure')
legend('R=0.1' , 'R=0.2','R=0.7' , 'R=0.9')
set(gca,'Xscale','log')

% lambda=9.0
r1=linspace(0.1,100,100000);
p1=1+4/3*1./r1*1/0.1^2*(1/4*(1-4*0.1^3)+3/4*9*(1-3*0.1^2))-0.1./
(3*r1.^4)*(1-0.1^3+3*9*(1-0.1^2));
a1=max(p1);b1=min(p1);[argvalue1, argmax1] = max(p1)
r2=linspace(0.2,100,100000);
p2=1+4/3*1./r2*1/0.2^2*(1/4*(1-4*0.2^3)+3/4*9*(1-3*0.2^2))-0.2./
(3*r2.^4)*(1-0.2^3+3*9*(1-0.2^2));
a2=max(p2);b2=min(p2);[argvalue2, argmax2] = max(p2)
r3=linspace(0.7,100,100000);
p3=1+4/3*1./r3*1/0.7^2*(1/4*(1-4*0.7^3)+3/4*9*(1-3*0.7^2))-0.7./
(3*r3.^4)*(1-0.7^3+3*9*(1-0.7^2));
a3=max(p3);b3=min(p3);[argvalue3, argmax3] = max(p3)
r4=linspace(0.9,100,100000);
p4=1+4/3*1./r4*1/0.9^2*(1/4*(1-4*0.9^3)+3/4*9*(1-3*0.9^2))-0.9./
(3*r4.^4)*(1-0.9^3+3*9*(1-0.9^2));
a4=max(p4);b4=min(p4);[argvalue4, argmax4] = max(p4)
plot(r1,p1/(a1-b1),r2,p2/(a2-b2),'g',r3,p3/(a3-b3),'c',r4,p4/(a4-b4),'m')
xlabel('r'), ylabel('pressure')
legend('R=0.1' , 'R=0.2','R=0.7' , 'R=0.9')
set(gca,'Xscale','log')

% for same R=0.2
r2=linspace(0.2,100,100000);
p1=1+4/3*1./r2*1/0.2^2*1/4*(1-4*0.2^3)-0.2./(3*r2.^4)*(1-0.2^3);
a1=max(p1);b1=min(p1);[argvalue1, argmax1] = max(p1)
p2=1+4/3*1./r2*1/0.2^2*(1/4*(1-4*0.2^3)+3/4*0.2*(1-3*0.2^2))-0.2./
(3*r2.^4)*(1-0.2^3+3*0.2*(1-0.2^2));
a2=max(p2);b2=min(p2);[argvalue2, argmax2] = max(p2)
p3=1+4/3*1./r2*1/0.2^2*(1/4*(1-4*0.2^3)+3/4*9*(1-3*0.2^2))-0.2./
(3*r2.^4)*(1-0.2^3+3*9*(1-0.2^2));
a3=max(p3);b3=min(p3);[argvalue3, argmax3] = max(p3)
plot(r2,p1/(a1-b1),r2,p2/(a2-b2),'g',r2,p3/(a3-b3),'c')
```

```
xlabel('r'), ylabel('pressure')
legend('lambda=0' , 'lambda=0.2','lambda=9.0')
set(gca,'Xscale','log')

% for same R=0.7
r2=linspace(0.7,100,100000);
p1=1+4/3*1./r2*1/0.7^2*1/4*(1-4*0.7^3)-0.7./(3*r2.^4)*(1-0.7^3);
a1=max(p1);b1=min(p1);[argvalue1, argmax1] = max(p1)
p2=1+4/3*1./r2*1/0.7^2*(1/4*(1-4*0.7^3)+3/4*0.2*(1-3*0.7^2))-0.7./
(3*r2.^4)*(1-0.7^3+3*0.2*(1-0.7^2));
a2=max(p2);b2=min(p2);[argvalue2, argmax2] = max(p2)
p3=1+4/3*1./r2*1/0.7^2*(1/4*(1-4*0.7^3)+3/4*9*(1-3*0.7^2))-0.7./
(3*r2.^4)*(1-0.7^3+3*9*(1-0.7^2));
a3=max(p3);b3=min(p3);[argvalue3, argmax3] = max(p3)
plot(r2,p1/(a1-b1),r2,p2/(a2-b2),'g',r2,p3/(a3-b3),'c')
xlabel('r'), ylabel('pressure')
legend('lambda=0' , 'lambda=0.2','lambda=9.0')
set(gca,'Xscale','log')
```

2.3 Question 3

2.4 Q3Eulerii(t0,h,p,tmax)

```
function Q3Eulerii(t0,h,p,tmax)
% t0=starting time,],h=step, p=lambda, tmax=endpt of the range time
% to avoid the trivial solution, use series soln of x for small t
for the first step.
% e.g. Q3Eulerii(1*10^(-6),1*10^(-6),0,1)
t=t0;x=1-5*(1+2*p)*t0^2/4;R=zeros(1,floor(tmax/h+1));R(1,1)=1;
T=[0 (0:h:(tmax-h))+t0];i=2;P=R;
while (0<=x)&&(x<1)
    f=-5/2*(2/3*(1-x^1.2)+2*p*(1-x^0.8))^0.5;
    if t<=tmax
        R(1,i)=x^2.5;
        a=1/4*(1-4*R(1,i)^3)+3/4*p*(1-3*R(1,i)^2);b=1-R(1,i)^3+
        3*p*(1-R(1,i)^2);
        if (a<0)&&(R(1,i)>0)
            P(1,i)=1;
```

```
        elseif (a>=0)&&(R(1,i)>0)
            P(1,i)=1+(a^4/b)^(1/3)/R(1,i)^3;
        else
            P(1,i)=0;
        end
        x=x+h*f; t=t+h; i=i+1;
    else
        R(1,i)=x^2.5;
        a=1/4*(1-4*R(1,i)^3)+3/4*p*(1-3*R(1,i)^2);b=1-R(1,i)^3+
        3*p*(1-R(1,i)^2);
        if a<0
            P(1,i)=1;
        else
            P(1,i)=1+(a^4/b)^(1/3)/R(1,i)^3;
        end
        x=1;
    end
end
[argvalue, argmin] = min(R);
tc=t0+(argmin-2)*h
subplot(1,2,1)
plot(T,R)
xlabel('time'), ylabel('Radius')
legend('R(t)')
subplot(1,2,2)
plot(T,log(P))
xlabel('time'),ylabel('Scaled Pressure log(p_{max})')
legend('log(p_{max}(t))')
end
```

2.4.1 solvetc

```
function solvetc
% apply numerical integration to solve (6) for tc with lambda=0.
f=@(x) (2/3)^0.5*(sin(x)).^(2/3);
tc=integral(f,pi/2,pi)
end
```

2.5 Question 4

2.5.1 Q4

```
Q4EulerR(1*10^(-6),1*10^(-6),0,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),0.01,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),0.1,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),1,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),10,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),1000,1)
hold on
Q4EulerR(1*10^(-6),1*10^(-6),100000,1)
%
%
Q4EulerP(1*10^(-6),1*10^(-6),0,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),0.01,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),0.1,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),1,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),10,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),1000,1)
hold on
Q4EulerP(1*10^(-6),1*10^(-6),100000,1)
```

2.5.2 Q4EulerR(t0,h,p,tmax)

```
function Q4EulerR(t0,h,p,tmax)
% t0=starting time,],h=step, p=lambda, tmax=endpt of the range time
% to avoid the trivial solution, use series soln of x
for small t for the first step.
% e.g. Q4EulerR(1*10^(-6),1*10^(-6),0,1)
```

```
t=t0;x=1-5*(1+2*p)*t0^2/4;R=zeros(1,floor(tmax/h+1));R(1,1)=1;
T=[0 (0:h:(tmax-h))+t0]; i=2;
while (0<=x)&&(x<1)
    f=-5/2*(2/3*(1-x^1.2)+2*p*(1-x^0.8))^0.5;
    if t<=tmax
        R(1,i)=x^2.5;
        x=x+h*f; t=t+h; i=i+1;
    else
        R(1,i)=x^2.5;x=1;
    end
end
[argvalue, argmin] = min(R);
tc=t0+(argmin-2)*h
plot(T,R)
xlabel('time'), ylabel('Radius')
end
```

2.5.3 Q4EulerP(t0,h,p,tmax)

```
function Q4EulerP(t0,h,p,tmax)
% t0=starting time,],h=step, p=lambda, tmax=endpt of the range time
% to avoid the trivial solution, use series soln of x
for small t for the first step.
% e.g. Q4EulerP(1*10^(-6),1*10^(-6),0,1)
t=t0;x=1-5*(1+2*p)*t0^2/4;R=zeros(1,floor(tmax/h+1));R(1,1)=1;
T=[0 (0:h:(tmax-h))+t0]; i=2;P=R;
while (0<=x)&&(x<1)
    f=-5/2*(2/3*(1-x^1.2)+2*p*(1-x^0.8))^0.5;
    if t<=tmax
        R(1,i)=x^2.5;
        a=1/4*(1-4*R(1,i)^3)+3/4*p*(1-3*R(1,i)^2);b=1-R(1,i)^3+
        3*p*(1-R(1,i)^2);
        if (a<0)&&(R(1,i)>0)
            P(1,i)=1;
        elseif (a>=0)&&(R(1,i)>0)
            P(1,i)=1+(a^4/b)^(1/3)/R(1,i)^3;
        else
            P(1,i)=0;
        end
```

```
        x=x+h*f;t=t+h;i=i+1;
    else
        R(1,i)=x^2.5;
        a=1/4*(1-4*R(1,i)^3)+3/4*p*(1-3*R(1,i)^2);b=1-R(1,i)^3+
        3*p*(1-R(1,i)^2);
        if a<0
            P(1,i)=1;
        else
            P(1,i)=1+(a^4/b)^(1/3)/R(1,i)^3;
        end
        x=1;
    end
end
plot(T,log(P))
xlabel('time'),ylabel('Scaled Pressure log(p_{max})')
legend('log(p_{max}(t))')
end
```

2.5.4 Q4plot

```
X=[0.01 0.1 1 10 1000 100000];
Y=[0.904763 0.828006 0.511718 0.191102 0.019545 0.001959];
plot(log(X),Y)
hold on
x=linspace(0.01,100000,3000000);
y=2./(5*(1+2*x)).^0.5;
plot(log(x),y)
xlabel('log(\lambda)'),ylabel('t_{c}')
```