The differential equations for chemical reaction kinetics are frequently highly stiff, partly because the coefficients in the models can vary by many orders of magnitude, and partly because of strong nonlinearities. Consider the simple combustion model* given in dimensionless form by

$$\dot{r} = r^2 - r^3, \tag{1}$$

in which the variable $r(t)$ represents the radius of a flame. Equation (1) models how the rate of growth of the size of a flame involves a competition between oxygen coming in through the surface of the flame ($\propto r^2$) and that being consumed inside the flame's volume ($\propto r^3$). By factoring Eq. (1) into the form $\dot{r} = r^2(1-r)$, we see that the flame will grow very rapidly from its initiation (when $0 < r \ll 1$), but will stop growing when the normalized radius $r = 1$. Furthermore, this steady state flame size is stable, since $\dot{r} > 0$ when $r < 1$ and $\dot{r} < 0$ when $r > 1$. So, this simple model explains why a candle, once lit, reaches a stable maximum size.

(a) In what follows, you are to solve Eq. (1) numerically for $r(0) = 0.01$ when $t = 0$, and for a final time of $t_f = 300$. First, use the explicit Euler method implemented previously for Problem 2 of Homework 4. Obtain and plot results (r vs. t) for step sizes $h = 0.1, 1, 2, 3$. What do you observe?

(b) Next, use an *implicit* Euler method defined by

$$r_{k+1} = r_k + f(t_{k+1}, r_{k+1})\,h \quad \Rightarrow \quad r_{k+1} - f(t_{k+1}, r_{k+1})\,h - r_k = 0\,. \tag{2}$$

At each step, it is necessary to use a root finder to solve for r_{k+1}. You may employ Matlab's built-in `fzero` command to do this by using

```
r(k+1)=fzero(@(R) R-(R^2-R^3)*h-r(k),r(k));
```
(3)

inside your main iteration loop, where `k` is the step. **Note:** The call to `fzero` shown in Eq. (3) finds the zeros of the anonymous function in `R` defined in its first argument. By comparing Eq. (3) with Eq. (2), you can see that the dummy variable `R` is actually the value of r_{k+1} to be found. The function takes the step size `h` and the value of r at the current step, `r(k)`, as known values. In Eq. (3), the second argument to `fzero` is the initial guess, which is taken to be `r(k)`. The resulting zero is then assigned to `r(k+1)`. Run your program using the same h values as in part (a), plot the results, and again describe what you observe.

*Taken from the Scholarpedia article "Stiff Systems" by L.F. Shampine and S. Thompson. Also, see Moler Ch. 7.9, for additional discussion.