Mario L. Gutierrez Abed
364009832
mlg3843@rit.edu

Problem Set 2
Math Modeling II

02-13-2021

Problem 1. As explained in class, there is a long standing debate within the literature about whether the metabolic function, $E_{metabolism} = \beta M^{\alpha}$, should be $E_{metabolism} = \beta_{2/3} M^{2/3}$ instead of $E_{metabolism} = \beta_{3/4} M^{3/4}$. In this first part of the homework, you will explore the implications of picking $\alpha = 2/3$ instead of $\alpha = 3/4$ in our proposed weight model given by

$$\frac{dM}{dt} = \lambda - \tilde{\beta}(\alpha) M^{\alpha}, \quad \text{where} \quad \lambda = \frac{E_{food} - E_{exercise}}{\rho} > 0, \quad \tilde{\beta} = \frac{\beta(\alpha)}{\rho}, \tag{1}$$

and $M(0) = M_{initial}$.

a) First find the equilibrium point, $\overline{M}_{2/3}$, of Eqn. (1) when $\alpha = 2/3$. Second, use linear stability analysis to determine the stability of Eqn. (1) near the equilibrium point $\overline{M}_{2/3}$. Next, comment on the global stability of the equilibrium point $\overline{M}_{2/3}$. Finally, thinking back to the application of dieting, if you wanted to construct a diet plan to achieve a final weight of M_{final}, comment on how your diet plan (determined by the value of λ) changes with changes in α.

b) Nondimensionalize Eqn. (1) to find the characteristic time scale, t_c, of weight change when $\alpha = 2/3$. To do so, assume the characteristic mass scale is given by $M_{initial}$; that is, $M = M_{initial}\overline{\overline{M}}$, where $\overline{\overline{M}}$ is the nondimensional mass. Thinking back to the application of dieting, comment on how the characteristic time scale changes as α changes.

c) Simulate multiple diet plans for one individual. Write your own code, in the language you feel most comfortable with, to approximate the solution to Eq. (1) for both $\alpha = 2/3$ and $\alpha = 3/4$. To help prepare you for the practical numerical analysis qualification exam, you can't use any packages. You must write your own algorithm. Please submit your code with this assignment. As done in class, assume the individual's initial weight, $M_{initial}$, is 275 lbs. The individual is interested in losing 10 lbs, $M_{final} = 265$ lbs. Create a summary of your findings. Your summary must include (i) justifications for parameter choices, (how did you pick ρ? how did you pick β for a given α?), (ii) a comparison of results from the two different models, and (iii) any thoughts you might have on whether one model is better than the other.

Solution to a). The equilibrium point, $\overline{M}$, [1] is reached when Eqn. (1) reaches a steady state; that is, when

$$\frac{dM}{dt} = 0 = \lambda - \tilde{\beta}\,\overline{M}^{2/3} \quad \Longrightarrow \quad \overline{M} = \left(\frac{\lambda}{\tilde{\beta}}\right)^{3/2}.$$

Now, similar to how we did in class, we consider a perturbation ε at the equilibrium point,

$$M(t) = \overline{M} + \varepsilon(t). \tag{2}$$

We assume that this perturbation is initially very small (i.e., $\varepsilon(0) \ll \overline{M}$), and then we check for (linear) stability by making sure that upon performing a (linear) expansion the perturbation dies out after some time. Plugging Eq. (2) into Eq. (1) and expanding,

$$\begin{aligned}
\frac{dM}{dt} &= \frac{d}{dt}\left(\overline{M} + \varepsilon(t)\right) \\
&= \lambda - \tilde{\beta}\left(\overline{M} + \varepsilon\right)^{2/3} && \text{(By Eq. (1))} \\
&\approx \lambda - \tilde{\beta}\left(\overline{M}^{2/3} + \frac{2}{3}\overline{M}^{-1/3}\varepsilon + O(\varepsilon^2)\right). && (3)
\end{aligned}$$

[1] To make the notation more readable, from now on I'm dropping all the references to $\alpha = 2/3$; it is understood which value of α we will be working with, depending on the particular question being tackled.

Now,

$$\frac{d}{dt}\left(\overline{M}+\varepsilon(t)\right) = \overbrace{\frac{d\overline{M}}{dt}}^{=0} + \frac{d\varepsilon}{dt} = \frac{d\varepsilon}{dt},$$

and

$$\lambda - \tilde{\beta}\,\overline{M}^{2/3} \xlongequal{\text{From Eq. (1)}} \frac{d\overline{M}}{dt} \xlongequal{\text{Steady State}} 0.$$

Thus, plugging back into Eq. (3) and ignoring higher-order ε terms, we end up with

$$\frac{d\varepsilon}{dt} = -\frac{2}{3}\tilde{\beta}\,\overline{M}^{-1/3}\varepsilon.$$

The quantities $\tilde{\beta}$, $\overline{M}$, and ε are all positive, so the minus sign in this result indicates an ever-decreasing perturbation ε; that is,

$$\lim_{t\to\infty}\varepsilon(t) = 0.$$

This demonstrates the (linear) stability of Eqn. (1) near the equilibrium point $\overline{M}$, which is what we wanted to show. This stability of the equilibrium point is, in fact, global, since any weight values away from this point will always be decreasing (or increasing) until they reach this steady-state value.

Finally, thinking back to the application of dieting, we see that our diet plans are sensitive to the value of α in the sense that for a fixed λ (value that we chose in order to reach our goal weight M_{final}), we may end up with a higher or lower M_{final} depending on the specific value of α. For instance, for $\alpha = 2/3$, for a fixed λ we end up with a higher final weight M_{final} than with a value of $\alpha = 3/4$. This can be easily seen from the expression of the equilibirum weight

$$\overline{M} = M_{final} = \left(\frac{\lambda}{\tilde{\beta}}\right)^{1/\alpha}. \qquad \square$$

Solution to b). As instructed, we assume that $M = M_{initial}\overline{\overline{M}}$ and, similarly, $t = t_c\overline{\overline{t}}$, where t_c is the characteristic time scale that we are looking for and $\overline{\overline{t}}$ denotes nondimensional time. Then we plug this into Eq. (1) and expand:

$$\begin{aligned}
\frac{dM}{dt} &= \lambda - \tilde{\beta}M^{2/3}\\
\frac{d\left(M_{initial}\overline{\overline{M}}\right)}{d\left(t_c\overline{\overline{t}}\right)} &= \lambda - \tilde{\beta}\left(M_{initial}\overline{\overline{M}}\right)^{2/3}\\
\frac{M_{initial}}{t_c}\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} &= \lambda - \tilde{\beta}\left(M_{initial}\overline{\overline{M}}\right)^{2/3}\\
\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} &= \frac{\lambda t_c}{M_{initial}} - \frac{\tilde{\beta}t_c}{M_{initial}^{1/3}}\overline{\overline{M}}^{2/3}.
\end{aligned} \tag{4}$$

At this point if we nondimensionalize the coefficient multiplying the $\overline{\overline{M}}$ factor,

$$\frac{\tilde{\beta}t_c}{M_{initial}^{1/3}} = 1,$$

we get an expression for the characteristic time scale,

$$t_c = \frac{M_{initial}^{1/3}}{\tilde{\beta}}. \tag{5}$$

Moreover, plugging this back into Eq. (4), we get

$$\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} = \frac{\lambda}{\tilde{\beta}\,M_{initial}^{2/3}} - \overline{\overline{M}}^{2/3}. \tag{4†}$$

Finally, from comparing Eq. (5) with the characteristic time corresponding to $\alpha = 3/4$,

$$ {}^{(3/4)}t_c = \frac{M_{initial}^{1/4}}{{}^{(3/4)}\tilde{\beta}}, $$

we can conclude that the characteristic time scale has an inversely proportional relation with α. $\square$

Solution to c). We now show the time evolution (over the course of 2000 days) of weight change for a woman that starts at the initial weight $M_{initial} = 275$ lbs and whose goal weight is $M_{final} = 265$ lbs. We simulate using both the $2/3$- and $3/4$-laws. Given that we know what her goal weight is, from the steady state equation we get the following value for λ:

$$ \lambda = \tilde{\beta} M_{final}^{\alpha}, $$

where $\tilde{\beta}$, as defined earlier, is given by $\tilde{\beta} = \beta(\alpha)/\rho$. Thus we need to determine values for ρ and $\beta(\alpha)$. For the former I'm choosing $\rho = 7700$ kcal/kg, which appears to have empirical data backing it up (from references in the lecture slides to Kevin D. Hall's paper). As for $\beta(3/4)$, we also have data, this time from Max Kleiber's paper (also referenced on the lecture slides); since we are working with a woman (assumed to be ~ 30 years old and have "standard" (i.e., average) height), we set $\beta(3/4) = 65.8$. For $\beta(2/3)$, on the other hand, we do not have any data presented in the lectures, but nevertheless we can infer an estimate based on the relation

$$ \beta = \rho \frac{\lambda}{M_{final}^{\alpha}}. $$

This shows that $\beta(2/3)$ will be somewhat bigger than $\beta(3/4)$; let us then pick $\beta(2/3) = 66.5$.

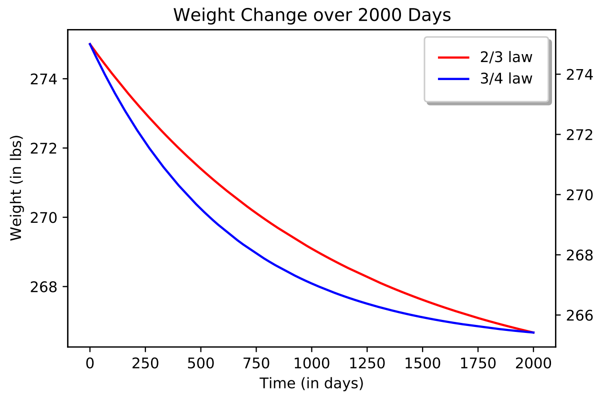
The following C++ *code implements the forward-Euler method to tackle Eq. (1):*

```
#include <fstream>
#include <cmath>
#include <vector>

using namespace std;

//Parameters
double rho {7700.0};
vector<double> alpha {2.0/3.0, 3.0/4.0};
vector<double> beta {66.5/rho, 65.8/rho};
double M_init {275.0};
double M_final {265.0};
vector<double> lmbd { beta.at(0) * pow(M_final, 2.0/3.0), beta.at(1)
                * pow(M_final, 3.0/4.0) };
double dt {1.0};
int max_days {2000};


int main(int argc, const char * argv[]) {

    ofstream myyfile ("weight.csv");  //OUTPUT WEIGHT DATA TO FILE
    vector<double> M {M_init, M_init};

    for (int i {0}; i <= max_days; i++){
        myyfile << M.at(0) << "," << M.at(1) << endl;
        M.at(0) = M.at(0) + dt * ( lmbd.at(0) - ( beta.at(0)
                    *  pow(M.at(0),alpha.at(0)) ) );
        M.at(1) = M.at(1) + dt * ( lmbd.at(1) - ( beta.at(1)
                    *  pow(M.at(1),alpha.at(1)) ) );
    }
    myyfile.close();

   return 0;
}
```

We now import this data into Python for plotting:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

font = {'family' : 'serif',
        'weight' : 'normal',
        'size'   : 14}

# Create the figure and axes
fig, ax1 = plt.subplots(1, 1)
ax2 = ax1.twinx()

weight = pd.read_csv("~/MyXCodeProjects/MathModeling2/Diet_Model/weight.csv",
         header = None)
max_days = 2000;
t = np.linspace(0, max_days, num=2001)

# Add labels for the legend
function1 = ax1.plot(t, weight[0], 'r', label=r'$2/3$ law')
function2 = ax2.plot(t, weight[1], 'b', label=r'$3/4$ law')

# Create the legend by first fetching the labels from the functions
functions = function1 + function2
labels = [f.get_label() for f in functions]
plt.legend(functions, labels, fancybox=True, framealpha=1, borderpad=1,
           shadow=True,  loc=0)

# Add labels and title
ax1.set_xlabel(r'Time (in days)')
ax1.set_ylabel(r'Weight (in lbs)')
plt.title('Weight Change over 2000 Days')

# Save the figure
plt.savefig('Figures/Weight.pdf', bbox_inches='tight')
plt.close()
```

As the figure indicates, the results are not very satisfactory using either α-law. It takes way too long to reach the goal weight, considering that the individual only wanted to shred 10 lbs.



Using

```cpp
cout << M.at(0)  << ", " << M.at(1) << "\n " <<  endl;
if ( (M.at(0) - M_final <= 0.5) ||  ((M.at(1) - M_final <= 0.5))  ) {
    cout << "The goal weight of " << M_final << " lbs has been reached after "
         << i << " days." << endl;
    break;
}
```

I did find that that the $3/4$-law gets within 0.5 of the desired weight M_{final} after $t = 1887$ days, while the $2/3$-law does not reach this weight even after all 2000 days, but that is not saying much ... Both methods are ineficient with the parameters that we started with; we need a more strict λ (i.e., less calories consumed/more calories burned daily). □

Problem 2. Let $M(t)$ denote the body mass of an individual at time t. We explore the consequences of modeling the body metabolism using the function $E_{metabolism} = \frac{AM}{1+BM^{1-\alpha}}$, where A and B are positive parameters, and $0 < \alpha < 1$. Now, the rate of change of M, with respect to time, is given by

$$\frac{dM}{dt} = \lambda - \frac{\tilde{A}M}{1+BM^{1-\alpha}}, \quad \text{where} \quad \lambda = \frac{E_{food} - E_{exercise}}{\rho} > 0, \quad \tilde{A} = \frac{A}{\rho}, \tag{6}$$

and $M(0) = M_{initial}$.

a) Determine the dimensions of the parameters A and B so that $E_{metabolism}$ has dimensions of energy/time.

b) Prove that Eq. (6) has exactly one positive real equilibrium point, $\overline{M}$. (Hint: Use ideas from first semester calculus.) Thinking back to the application of dieting, what does one positive real equilibrium point imply? What if there were two positive real equilibrium points? What would that imply about your weight change?

c) Use linear stability analysis to determine the stability of the Eqn. (6) near the positive real equilibrium point $\overline{M}$. Next, comment on the global stability of the equilibrium point $\overline{M}$.

d) Nondimensionalize Eqn. (6) to find the characterize time scale, t_c, of weight change. Again, to do so, assume the characteristic mass scale is $M_{initial}$; that is, $M = M_{initial}\overline{\overline{M}}$, where $\overline{\overline{M}}$ is the nondimensional mass. How does this characteristic time scale compare to the characterize time scale found in #1?

e) Finally, simulate multiple diet plans for one individual. To do so, modify the code you wrote for question #1. As above, assume the individual's initial weight, $M_{initial}$, is 275 lbs. The individual is interested in losing 10 lbs, $M_{final} = 265$ lbs. Create a summary of your findings. Your summary must include (i) justifications for parameter choices, (how did you pick ρ? how did you pick A, B, and α?), (ii) a comparison of results from Eqn. (1) and (6), and (iii) any thoughts you might have on whether one model is better than the other.

Solution to a). From

$$[E_{metabolism}] = \left[\frac{AM}{1+BM^{1-\alpha}}\right],$$

we have

$$\frac{E}{T} = \frac{[A]\,M}{[B]\,M^{1-\alpha}} = M^{\alpha}\frac{[A]}{[B]}.$$

In order for this equality to hold, we conclude that B has time units, and A has units energy $\times$ mass$^{-\alpha}$. □

Solution to b). The equilibrium point, $\overline{M}$, is found when Eq. (6) reaches a steady state, i.e., when dM/dt vanishes, or, equivalently, when

$$\lambda = \frac{\tilde{A}\overline{M}}{1+B\overline{M}^{1-\alpha}}.$$

If we define the function

$$\Psi(M) \equiv \lambda - \frac{\tilde{A}M}{1+BM^{1-\alpha}},$$

then the equilibrium point, $\overline{M}$, is precisely the root of Ψ. We just need to show that this root is unique; to do so let us consider how Ψ behaves as we change weight. In other words, consider the derivative

$$\Psi'(M) = -\frac{(\tilde{A}M)'\left(1+BM^{1-\alpha}\right) - \tilde{A}M\left(1+BM^{1-\alpha}\right)'}{\left(1+BM^{1-\alpha}\right)^2},$$

where we fixed λ and used the quotient rule from elementary calculus. Expanding the numerator on the RHS and simplifying, we get

$$\Psi'(M) = -\frac{\tilde{A}\left(1+\alpha B M^{1-\alpha}\right)}{\left(1+BM^{1-\alpha}\right)^2}.$$

Since the fraction is necessarily positive, the derivative is always negative, which in turn implies that the function Ψ is ever-decreasing for all M (of course, M is assumed to be always positive). From this we deduce that Ψ can equal 0 once and only once, thus the root $\overline{M}$ is unique.

In the derivation above we assumed a fixed λ (fixed diet plan). Had we found instead that there were multiple equilibria, then for a chosen diet plan we would have different target weights, which would make the model practically useless. We want our model to be such that for a given diet plan (fixed λ) we can aim to reach a specific desired weight. ◻

Solution to c). *We consider a linear perturbation ε at the equilibrium point,*

$$M(t) = \overline{M} + \varepsilon(t). \tag{7}$$

We insert this expression this time into Eq. (6) and expand,

$$\begin{aligned}
\frac{dM}{dt} &= \frac{d}{dt}\left(\overline{M} + \varepsilon(t)\right) \\
&= \lambda - \frac{\tilde{A}\left(\overline{M}+\varepsilon\right)}{1+B\left(\overline{M}+\varepsilon\right)^{1-\alpha}} && \text{(By Eq. (6))} \\
&\approx \lambda - \frac{\tilde{A}\left(\overline{M}+\overline{M}^{-1}\varepsilon + O\left(\varepsilon^2\right)\right)}{1+B\left(\overline{M}^{1-\alpha}+(1-\alpha)\overline{M}^{-\alpha}\varepsilon + O\left(\varepsilon^2\right)\right)}.
\end{aligned} \tag{8}$$

This one is much harder to crack than the previous one ... Recall, as previously, that

$$\frac{d}{dt}\left(\overline{M} + \varepsilon(t)\right) = \frac{d\varepsilon}{dt}.$$

Then, considering only terms up to first oder, let's rewrite (8) as

$$\frac{d\varepsilon}{dt} = \lambda - \underbrace{\frac{\tilde{A}\overline{M}}{1+B\left(\overline{M}^{1-\alpha}+(1-\alpha)\overline{M}^{-\alpha}\varepsilon\right)}}_{(\dagger)} - \underbrace{\frac{\tilde{A}\overline{M}^{-1}\varepsilon}{1+B\left(\overline{M}^{1-\alpha}+(1-\alpha)\overline{M}^{-\alpha}\varepsilon\right)}}_{(\ddagger)}.$$

Recalling that, at the steady state,

$$\lambda = \frac{\tilde{A}\overline{M}}{1+B\overline{M}^{1-\alpha}},$$

we can easily see that $0 < (\dagger) < 1$. If we could now only show that $(\ddagger) > (\dagger)$, then it would follow that ε is an ever-decreasing perturbation; i.e.,

$$\lim_{t\to\infty} \varepsilon(t) = 0.$$

However, proving this final step has taken me to some very messy algebra and I just haven't been able to figure it out just yet. This stability would be global, since any weight values away from this point will always be decreasing (or increasing) until they reach this steady-state value. ◻

Solution to d). Plugging into Eq. (6) and expanding,

$$\begin{aligned}
\frac{dM}{dt} &= \lambda - \frac{\tilde{A}M}{1 + BM^{1-\alpha}} \\
\frac{d\left(M_{initial}\overline{\overline{M}}\right)}{d\left(t_c\overline{\overline{t}}\right)} &= \lambda - \frac{\tilde{A}\,M_{initial}\overline{\overline{M}}}{1 + B\left(M_{initial}\overline{\overline{M}}\right)^{1-\alpha}} \\
\frac{M_{initial}}{t_c}\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} &= \lambda - \frac{\tilde{A}\,M_{initial}\overline{\overline{M}}}{1 + BM_{initial}^{1-\alpha}\,\overline{\overline{M}}^{1-\alpha}} \\
\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} &= \frac{\lambda t_c}{M_{initial}} - \frac{\tilde{A}\overline{\overline{M}}\,t_c}{1 + BM_{initial}^{1-\alpha}\,\overline{\overline{M}}^{1-\alpha}}.
\end{aligned} \tag{9}$$

This time the coefficient multiplying the $\overline{\overline{M}}$ factor is far more complicated than the λ factor, so we may choose instead

$$\frac{\lambda t_c}{M_{initial}} = 1 \quad \Longrightarrow \quad t_c = \frac{M_{initial}}{\lambda},$$

Plugging this back into Eq. (9), we get

$$\frac{d\overline{\overline{M}}}{d\overline{\overline{t}}} = 1 - \frac{\tilde{A}\,M_{initial}\,\overline{\overline{M}}}{\lambda\left[1 + BM_{initial}^{1-\alpha}\,\overline{\overline{M}}^{1-\alpha}\right]}. \tag{11†}$$

On Problem 1 we concluded that the characteristic time scale and α had an inversely proportional relation. This time with the characteristic time we chose there is no (direct) relation to α, although we can clearly see from Eq. (11†) that the larger α is the smaller the denominator on the second expression of the equation, which would then lead to a sharper decrease in weight over time. □

Solution to e). We now show the time evolution of the weight change given by Eq. (6). Once again our individual[2] has an initial weight $M_{initial} = 275$ lbs and wants to reach $M_{final} = 265$ lbs. We will run the simulation for four different values for α ($\{1/2,\ 2/3,\ 3/4,\ 4/5\}$). Since we know what the goal weight M_{final} is, from the steady state equation we get the following value for λ:

$$\lambda = \frac{\tilde{A}M_{final}}{1 + BM_{final}^{1-\alpha}}.$$

Thus we need to determine values for $\tilde{A}$ and B. From the dimensional analysis on part a), we know that A has units energy $\times$ mass$^{-\alpha}$; therefore $\tilde{A}$ must have units mass$^{1-\alpha}$. Not having any empirical data in hand, we cannot know exactly the proportionality, but a somewhat reasonable assumption would be $\tilde{A} := M_{initial}^{1-\alpha}$. Let's see where that leads us ... As for B, since it has time units, a sound assumption/simplification would be to set it $B := \Delta t = 1$.

The following C++ code implements the forward-Euler method to tackle Eq. (6):

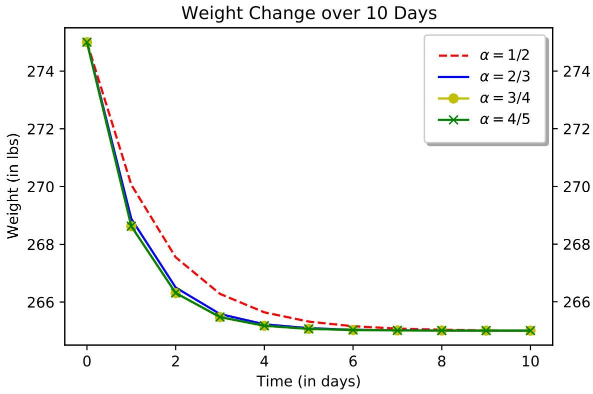
```
#include <fstream>
#include <cmath>
#include <vector>

using namespace std;

//Parameters
int max_days {10};
double dt {1.0};
double B {1.0};
double M_init {275.0};
double M_final {265.0};
vector<double> alpha {1.0/2.0, 2.0/3.0, 3.0/4.0, 4.0/5.0};
```

[2] This time we do not take into consideration whether the person is male or female, since we lack empirical data for this new model.

```cpp
vector<double> A { pow(265, (1-alpha.at(0)) ), pow(265, (1-alpha.at(1)) ),
                   pow(265, (1-alpha.at(2)) ), pow(265, (1-alpha.at(3)) ) };

vector<double> lmbd {
                   (A.at(0) * M_final)/(1.0 + B * pow(M_final, (1.0 - alpha.at(0)) ) ),
                   (A.at(1) * M_final)/(1.0 + B * pow(M_final, (1.0 - alpha.at(1)) ) ),
                   (A.at(2) * M_final)/(1.0 + B * pow(M_final, (1.0 - alpha.at(2)) ) ),
                   (A.at(3) * M_final)/(1.0 + B * pow(M_final, (1.0 - alpha.at(3)) ) )
                   };

int main(int argc, const char * argv[]) {

    ofstream myyfile ("weight_second_model.csv");  //OUTPUT WEIGHT DATA TO FILE
    vector<double> M (4, M_init);

    for (int i {0}; i <= max_days; i++){
        myyfile << M.at(0) << "," << M.at(1)  << "," << M.at(2)
                << "," << M.at(3) << endl;

        M.at(0) = M.at(0) + dt * ( lmbd.at(0) - ( (A.at(0) * M.at(0))/
                  ( 1.0 + B * pow(M.at(0), (1-alpha.at(0))) ) ) );
        M.at(1) = M.at(1) + dt * ( lmbd.at(1) - ( (A.at(1) * M.at(1))/
                  ( 1.0 + B * pow(M.at(1), (1-alpha.at(1))) ) ) );
        M.at(2) = M.at(2) + dt * ( lmbd.at(2) - ( (A.at(2) * M.at(2))/
                  ( 1.0 + B * pow(M.at(2), (1-alpha.at(2))) ) ) );
        M.at(3) = M.at(3) + dt * ( lmbd.at(3) - ( (A.at(3) * M.at(3))/
                  ( 1.0 + B * pow(M.at(3), (1-alpha.at(3))) ) ) );
    }
    myyfile.close();
return 0;
}
```

This time we only ran the code for 10(!) days. Yes, the model is much more efficient than the one we encountered in Problem 1! ☺ Let us import the data into Python and plot:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

font = {'family' : 'serif',
        'weight' : 'normal',
        'size'   : 14}

max_days = 10;

# Create the figure and axes
fig, ax1 = plt.subplots(1, 1)
ax2 = ax1.twinx()

new_weight = pd.read_csv("~/MyXCodeProjects/MathModeling2/Diet_MoreComplex/
    weight_second_model.csv", header = None)
t = np.linspace(0, max_days, num=11)

# Add labels for the legend
function1 = ax1.plot(t, new_weight[0], 'r--', label=r'$\alpha = 1/2$')
function2 = ax2.plot(t, new_weight[1], 'b-', label=r'$\alpha = 2/3$')
function3 = ax2.plot(t, new_weight[2], 'y-o', label=r'$\alpha = 3/4$')
function4 = ax2.plot(t, new_weight[3], 'g-x', label=r'$\alpha = 4/5$')

# Create the legend by first fetching the labels from the functions
functions = function1 + function2 + function3 + function4
labels = [f.get_label() for f in functions]
plt.legend(functions, labels, fancybox=True, framealpha=1, borderpad=1, shadow=True,
    loc=0)
# ax1.set_xlim(0, 10)

# Add labels and title
ax1.set_xlabel(r'Time (in days)')
ax1.set_ylabel(r'Weight (in lbs)')
plt.title('Weight Change over 10 Days')

# Save the figure
plt.savefig('Figures/Weight_newModel.pdf', bbox_inches='tight')
plt.close()
```



The results show a much more satisfactory model than the previous one. Of course, one may argue that it is not very realistic (or at the very least not very healthy!) to lose this much weight in so few days; indeed, perhaps some of the assumptions made are not entirely feasible and some parameters need to be fine-tuned . . . But at least superficially, this does look a much better model for someone trying lose some weight in a reasonable time span. □