Lecture 1: Newton Cooling and Stability

Introduction. Many quantities change as time progresses such as money in a savings account or the temperature of a refreshing drink or any cooling mass. Here we will be interested in making predictions about such changing quantities. A simple mathematical model has the form $\text{new } y = a \times \text{old } y + b$ where $a$ and $b$ are given real numbers. This calculation is usually repeated a number of times and is an example of an algorithm. Because of the large number of repeated calculations, we usually use some computing tool.

Consider the cooling of a well-stirred liquid such as a cup of coffee. Here we want to predict the temperature of the liquid given on some initial observations. Newton’s law of cooling is based on the observation that for small changes of time, $h$, the change in the temperature is nearly equal to the product of the some positive constant $c$, the $h$ and the difference in the room temperature and the present temperature of the coffee. Consider the following quantities:

- $y^k$ equals the temperature of a well-stirred cup of coffee at time $t_k$,
- $y_{sur}$ equals the surrounding room temperature, and
- $c$ measures the insulation ability of the cup.

The symbolic form of the discrete Newton's law of cooling is

$$y^{k+1} - y^k = ch(y_{sur} - y^k).$$

Or,

$$y^{k+1} = (1 - ch) y^k + chy_{sur}$$

$$= a y^k + b$$

where $a = 1 - ch$ and $b = chy_{sur}$. The long run solution should be the room temperature, that is, $y^k$ should converge monotonically to $y_{sur}$ as $k$ increases. This does happen if we impose the $1 - ch > 0$ condition on the time step $h$; this is often called a stability condition on $h$.

Model. The model in this case appears to be very simple. It consists of three constants $y^0$, $a$, $b$ and the formula

$$y^{k+1} = a y^k + b. \hspace{2cm} (1)$$

The formula must be used repeatedly, but with different $y^k$ being put into the right side. Often $a$ and $b$ are derived from formulating how $y^k$ changes as $k$ increases ($k$ reflects the time step). The change in the amount $y^k$ is often modeled by $dy^k + b$

$$y^{k+1} - y^k = d y^k + b$$
where \( a = 1 + d \). This model given in (1) is called a **first order finite difference** model for the sequence of numbers \( y^{k+1} \). Later we will generalize this to a sequence of column vectors where the single number, \( a \), will be replaced by a square matrix.

**Method.** The "iterative or recursive" calculation of (1) is the most common approach to solving (1). For example, if \( a = \frac{1}{2}, b = 2 \) and \( y^0 = 10 \), then
\[
\begin{align*}
y^1 &= \frac{1}{2} \cdot 10 + 2 = 7, \\
y^2 &= \frac{1}{2} \cdot 7 + 2 = 5.5, \\
y^3 &= \frac{1}{2} \cdot 5.5 + 2 = 4.75
\end{align*}
\]
.........
If one needs to compute \( y^{k+1} \) for large \( k \), this can get a little tiresome. Furthermore, if the calculations are being done with a computer, then the floating point errors may begin to have significant accumulation errors.

An alternative method is to use the following "telescoping" calculation and the geometric summation. Recall the **geometric summation**, \( 1 + r + r^2 + \ldots + r^k \), and
\[
(1 + r + r^2 + \ldots + r^k)(1 - r) = 1 - r^{k+1}.
\]
Or, for \( r \) not equal to 1
\[
(1 + r + r^2 + \ldots + r^k) = (1 - r^{k+1})/(1 - r).
\]
Consequently, if \( |r| < 1 \), then \( 1 + r + r^2 + \ldots = 1/(1 - r) \) is the **convergent geometric series**.

In (1) we can compute \( y^k \) by decreasing \( k \) by 1 so that \( y^k = a y^{k-1} + b \). Put this into (1) to get
\[
y^{k+1} = a (a y^{k-1} + b) + b
\]
\[
= a^2 y^{k-1} + a b + b
\]
\[
= a^2 (a y^{k-2} + b) + a b + b
\]
\[
= a^3 y^{k-2} + a^2 b + a b + b
\]
\[
\vdots
\]
\[
= a^{k+1} y^0 + b (a^k + \ldots + a^2 + a + 1)
\]
\[
= a^{k+1} y^0 + b (1 - a^{k+1})/(1 - a).
\]

**Steady State Theorem.** If \( a \) is not equal to 1, then the solution of (1) has the form given in (2). Moreover, if \( |a| < 1 \), then the solution of (1) will converge to the steady state solution \( y = ay + b \), that is, \( y = b/(1-a) \). More precisely, the error is given by
\[
y^{k+1} - y = a^{k+1} (y^0 - b/(1 - a)).
\]
The error for the steady state solution will be small if $|a|$ is small, or $k$ is large, or the initial guess $y^0$ is close to the steady state solution $b/(1 - a)$.

**Implementation.** The application to heat transfer is as follows. Consider a cup of coffee, which is initially at 200 degrees and is in a room with temperature equal to 70, and after 5 minutes it cools to 190 degrees. By using $h = 5$, $y^0 = 200$ and $y^1 = 190$, we compute from (1)

$$
y^1 = (1 - c h)y^0 + c h y_{sur}
$$

$$
190 = (1 - c 5)200 + (c 5) 70
$$

$$
190 - 200 = c(-1000 + 350)
$$

$$
-10 = c(-650)
$$

$$
c = 1/65.
$$

The following is a Matlab implementation of the above algorithm.

**Matlab Code in fofdh.m.**

```matlab
clear;
%Initial Time
t(1) = 0;
%Initial Temperature
y(1) = 200.;
h = 5;
% Number of Time Steps of Length h
n = 60;
% From y(2) = 190
a = 60/65;
b = 350/65;
% Execute the FOFD Algorithm
for k = 1:n
y(k+1) = a*y(k) + b;
t(k+1) = t(k) + h;
end
plot(t,y)
```

The first figure is for this $c$ and $h = 5$ so that

$$
a = 1 - c h = 60/65 \text{ and }
$$

$$
b = c h 70 = 350/65.
$$

The figure indicates the expected monotonic decrease to the steady state room temperature.
The next calculation is for a larger \( c = 2/13 \), which is computed from a new second observed temperature of 100 after 5 minutes. In this case for larger time step \( h = 10 \)

\[
a = 1 - (2/13)10 = -7/13 \text{ and } b = c h 70 = (2/13) 10 70 = 1400/13.
\]

Notice that the computed solution no longer is monotonic, but it does converge to the steady state solution! This indicates that our model for this large a time step is not very good.
The model continues to degrade as the magnitude of \( a \) increases. In the following figure the computed solution blows up!. This is consistent with formula (2). Here we kept the same \( c \), but let the step size increase to \( h = 16 \) and in this case

\[
\begin{align*}
a &= 1 - \frac{2}{13} \times 15 = -\frac{17}{13} \\
b &= c \times h = \frac{2}{13} \times 15 \times 70 = \frac{2100}{13}.
\end{align*}
\]

**Figure:** “Temperature” versus Time, \( a = -\frac{17}{13} \)

**Assessment.** In the case of the heat transfer problem, the formula for the temperature at the next time step is only an approximation, which gets better as the time step \( h \) decreases. The cooling process is continuous because the temperature changes at every instant in time. We have used a discretized model of this, and it seems to give good predictions provided the time step is suitably small. The discrete model was Euler’s method applied to the **continuous version of Newton’s Law of Cooling:**

\[
y'(t) = c(y_{\text{sur}} - y(t))
\]

In the next lecture we will more carefully consider the discretization error, which is defined to be the difference in the calculated values with no roundoff errors and the exact solution of the differential equation.
There are other modes of transferring heat such as diffusion and radiation. If the mass does not have uniform temperature, the heat will flow from the hotter regions to the cooler regions. Or, if the temperature is very high, then the mass may “glow”, that is, it will be cooled via emission of radiation. Both these require more complicated mathematical models, and the exact solution of the differential equations may not be explicitly known. In such cases we then use the discrete model and computers to do the tiresome calculations. However, as we have seen for Euler’s method, one must be careful to choose the parameters of the discrete model so as to obtain good approximations of the continuous models. Here mathematical analysis can be very helpful.

**Homework.**

1. Consider the application to Newton's discrete law of cooling. Use (2) to verify that if \( 1 - hc > 0 \), then \( y_{k+1} \) converges to the room temperature.

2. Write a computer code for the first order difference algorithm and apply it to a cooling cup of coffee whose initial temperature is 210, and after 2 minutes it has cooled to 205.

3. Consider a savings plan whose initial deposit is $1000, interest rate is 6% compounded monthly, and monthly deposit is $200.
   (a). Find a first order finite difference model for a savings plan.
   (b). Determine when this savings plan will have $20,000.