Summary

Newton's Law of Cooling/Heating

The rate of change \( \frac{dQ}{dt} \) of the temperature of an object is proportional to the difference between the temperature \( M(t) \) of the surrounding medium and the temperature \( Q(t) \) of the object itself; i.e., \( \frac{dQ}{dt} = K(M - Q) \), where \( K \) is a proportionality constant. In applications of this law, \( M \) is often constant.

Heating and Cooling of Buildings

A general model is \( \frac{dQ}{dt} = K(M - Q) \).

- \( M(t) \) is the temperature of the surrounding medium.
- \( Q(t) \) is the temperature inside the building.
- \( H(t) \) is the rate of increase in inside temperature due to people, lights, machines, etc.
- \( U(t) \) is the rate of increase/decrease in inside temperature due to heating/air conditioning, respectively.

The reciprocal \( 1/K \) is known as the time constant for the building.

Note on nomenclature

MATLAB on all platforms is case insensitive. That is, it internally regards \( T \) and \( t \) as different variables. The TI-89, like DOS or Windows, does not. Accordingly, I have chosen to use \( Q \) for \( T \) above, so as to pick something that will work on all platforms. (Of course, this is \( q \) on the ‘89.)

Hand/MATLAB Examples

107/4

A red wine is brought up from the wine cellar, which is a cool 10°C, and then left to breathe in a room of temperature 23°C. If it takes 10 minutes for the wine to reach 15°C, when will the temperature of the wine reach 18°C and be ready to drink?

Solution

Let \( Q(t) \) be the temperature of the wine \( t \) minutes after being brought up from the cellar. Newton's Law gives

\[
\frac{dQ}{dt} = K(23 - Q), \quad Q(0) = 10.
\]

Thus \( Q(t) = 23 - 13e^{-Kt} \). Using the 10-minute data, solve 15 = 23 - 13e^{-10K} to obtain \( K = -\frac{1}{10} \ln \frac{8}{13} \). Therefore, \( Q(t) = 23 - 13e^{\left(\frac{1}{10} \ln \frac{8}{13}\right)t} \). Finally, set \( Q = 18 \), then solve 18 = 23 - 13e^{\left(\frac{1}{10} \ln \frac{8}{13}\right)t} to get \( t = \frac{10 \ln \frac{5}{13}}{\ln \frac{8}{13}} \approx 19.68 \) minutes.

107/8

A garage with no heating or cooling has a time constant of \( 1/K = 2 \) hr. If the outside temperature varies as a sine wave with a minimum temperature of 50°F at 2:00 AM and a maximum of 80°F at 2:00 PM, determine the times at which the building reaches its lowest temperature and its highest temperature, assuming the exponential term has died off.

Solution

Let \( Q(t) \) be the temperature inside the garage and \( t \) be the number of hours past 8:00 AM. A little reflection shows that the outside
temperature is \( M(t) = 65 + 15 \sin(\pi t/12) \).

Accordingly, our IVP is \( \frac{dQ}{dt} = \frac{1}{2}(M(t) - Q(t)) \). \( Q(0) = 65 \). Using \texttt{dsolve} to obtain \( Q(t) \), we then plot it over a long enough time span to ensure that its exponential component has died off.

To find the time of day when the lowest garage temperature occurs, we symbolically compute \( \frac{dQ}{dt} \), then make an inline vectorized function out of it. Now numerically solve for \( \frac{dQ}{dt} = 0 \) near \( t = 68 \) (chosen by looking at the graph). After a little clock arithmetic, we ascertain that the minimum garage temperature is 51.7°F at 3:51 AM.

In a similar manner, we determine that the maximum garage temperature of 78.3°F occurs at 3:51 PM.

Note well that these extrema lag behind those of the outside temperature.

Here is the nitty-gritty. First the script M-file.

```matlab
% NSS4-107/8
syms K t
Q = dsolve('DQ = 1/2*(65+15*sin(pi*t/12) - Q)', ...
'Q(0)=65', 't') % temperature of building

pretty(Q)

dQ_dt = diff(Q,t); % derivative of temp 'o' building

t = linspace(0, 4*24, 1000);
Q = inline(vectorize(Q), 't');
plot(t,Q(t)); grid on

Q = 90*exp(-1/2*t)/(36+pi^2)*pi + 5*(468+13*pi-18*pi*cos(1/12*pi*t)+108*sin(1/12*pi*t))/(36+pi^2)

pretty(Q)

dQ_dt = diff(Q,t); % derivative of temp 'o' building

t = linspace(0, 4*24, 1000);
Q = inline(vectorize(Q), 't');
plot(t,Q(t)); grid on

Q = 90*exp(-1/2*t)/(36+pi^2)*pi + 5*(468+13*pi-18*pi*cos(1/12*pi*t)+108*sin(1/12*pi*t))/(36+pi^2)

low_time = fzero(dQ_dt, 68) % time when minimum occurs
high_time = fzero(dQ_dt, 80) % time when maximum occurs
low_temp = Q(low_time) % minimum building temperature
high_temp = Q(high_time) % maximum building temperature

% "Can you say clock arithmetic?" - Mr Rogers
LTOD = mod(8+low_time, 24) % time of day of low temp
LTOD_H = floor(LTOD) % the hour (military time)
LTOD_M = round(60*(LTOD - LTOD_H)) % the minute

HTOD = mod(8+high_time, 24) % time of day of high temp
HTOD_H = floor(HTOD) % the hour (military time)
HTOD_M = round(60*(HTOD - HTOD_H)) % the minute

echo off; diary off
```

And herewith the diary file.

```matlab
% NSS4-107/8
syms K t
Q = dsolve('DQ = 1/2*(65+15*sin(pi*t/12) - Q)', ...
'Q(0)=65', 't') % temperature of building

Q = 90*exp(-1/2*t)/(36+pi^2)*pi + 5*(468+13*pi-18*pi*cos(1/12*pi*t)+108*sin(1/12*pi*t))/(36+pi^2)

pretty(Q)

low_time = fzero(dQ_dt, 68) % time when minimum occurs
low_time = 67.8424
high_time = fzero(dQ_dt, 80) % time when maximum occurs
high_time = 79.8424
low_temp = Q(low_time) % minimum building temperature
low_temp = 51.7114
high_temp = Q(high_time) % maximum building temperature
high_temp = 78.2886

% "Can you say clock arithmetic?" - Mr Rogers
LTOD = mod(8+low_time, 24) % time of day of low temp
LTOD_H = 3
LTOD_M = round(60*(LTOD - LTOD_H)) % the minute
LTOD_M = 51
```
\[
\frac{dQ}{dt} = k \left( M^4 - Q^4 \right)
\]
\[
\frac{dQ}{dt} \approx k (M - Q) (M + Q) \left( M^2 + Q^2 \right)
\]
\[
\frac{dQ}{dt} \approx 4M^3k (M - Q)
\]
which is Newton’s law!

Finally, notice that if \( Q \) is close to \( M \) and \( M \) is constant, we have
\[
\frac{dQ}{dt} \approx k_1 (M - Q),
\]

Stefan’s law of radiation states that the rate of change of temperature of a body at \( Q \) degrees Kelvin in a medium at \( M \) degrees Kelvin is proportional to \( M^4 - Q^4 \). In other words, \( \frac{dQ}{dt} = k (M^4 - Q^4) \), where \( k \) is a positive constant. Solve this equation. Then explain why Newton’s law and Stefan’s law are nearly the same when \( Q \) is close to \( M \) and \( M \) is a constant.

**Solution**

We kick off the festivities with a massive MATLAB salvo.

```matlab
% NSS4-108/15
% impl_sol = dsolve('DQ = k*(M^4 - Q^4)', 't');
Warning: Explicit solution could not be found;
implicit solution returned.
pretty(impl_sol)
```

\[
\frac{\text{atan}(Q/M)}{3} - \frac{\log(Q + M)}{3} + \frac{\log(Q - M)}{3} + C1 = 0
\]

\[
\frac{t}{k} - \frac{1}{2} \frac{1}{k} M - 1/4 \frac{1}{k} M + 1/4 \frac{1}{k} M + C1 = 0
\]

```
% echo off; diary off
```

And now we hear the Sergeant cry, “Fix bayonets!”

\[
2 \tan^{-1} \left( \frac{Q/M}{4M^3} \right) + \ln(Q + M) - \ln(Q - M) = kt + kC_1
\]
\[
2 \tan^{-1} \left( \frac{Q/M}{4M^3} \right) + \ln(Q + M) - \ln(Q - M) = 4M^3kt + 4M^3kC_1
\]
\[
2 \tan^{-1} \left( \frac{Q/M}{4M^3} \right) - 4M^3kt - 4M^3kC_1 = \ln \left( \frac{Q - M}{Q + M} \right)
\]
\[
\exp \left( 2 \tan^{-1} \left( \frac{Q/M}{4M^3} \right) - 4M^3kt \right) e^{-4M^3kC_1} = \frac{Q - M}{Q + M}
\]

whence, setting \( C = e^{-4M^3kC_1} \), we have

\[
Q - M = C (Q + M) e^{2 \tan^{-1} \left( \frac{Q}{M} \right) - 4M^3kt},
\]

the answer the authors give (identifying \( Q \) with their \( T \)).