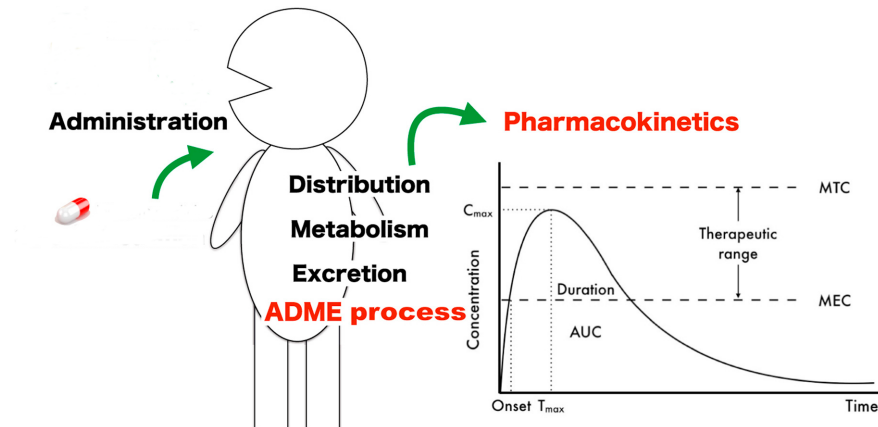


# MATLAB Demonstration

## MATH2601: Control of Partial Differential Equations



*Kill the disease, not the patient!*

Location: [http://people.sc.fsu.edu/~jburkardt/classes/control\\_2019/matlab\\_demo/matlab\\_demo.pdf](http://people.sc.fsu.edu/~jburkardt/classes/control_2019/matlab_demo/matlab_demo.pdf)

A good computing language enables us to turn our mathematical ideas into experimental mathematical tools. Let's see how MATLAB can help us take a simple model of medical drug dosages, and use it so estimate how to choose a treatment that delivers an effective level of medicine, without reaching a toxic level.

### MATLAB Demo

*How can we use a scientific programming language like MATLAB to:*

- *create a mathematical model of a real-world problem;*
- *devise an algorithm that can simulate the problem;*
- *write a computer program to implement the algorithm;*
- *experimentally search for solutions;*
- *graphically illustrate the properties of the solution;*

## 1 Modeling a Filter

Let's start by examining the process of *filtering*. For example, imagine we are proofreading a book for errors, and that we know that after a single reading, we will have caught and corrected just 50% of them. If the book contains 1,024 errors, then we will have reduced this to 512. A second pass would cut this to 256, a third to 128 and so on. Thus, the number of errors remaining in the book after  $n$  passes could be modeled by:

$$e(n) = e(0) \cdot \left(\frac{1}{2}\right)^n$$

If, instead of happening in discrete steps, the filtering is done continuously, then we might describe the number of errors as a function of time  $t$ :

$$e(t) = e(0) \cdot e^{\log(\frac{1}{2})t}$$

If we didn't see the formula for the number of errors, we could still express the fact that the errors are filtered at a certain rate by writing the differential equation:

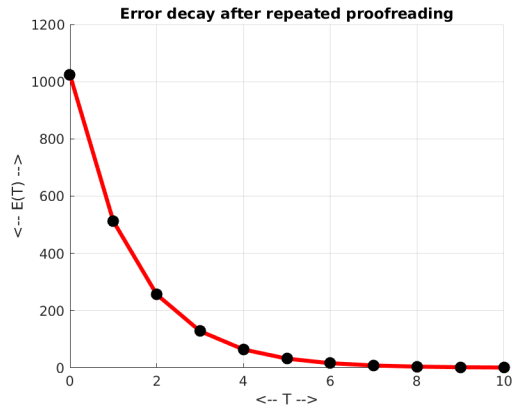
$$\frac{de}{dt} = \log\left(\frac{1}{2}\right) \cdot e(t)$$

MATLAB can illustrate this process:

```

1  n = 11;
2  c = log ( 0.5 );
3  t = linspace ( 0, 10, n );
4  e = 1024 * exp ( c * t );
5
6  hold ( 'on' );
7  plot ( t, e, 'r-', 'linewidth', 3 );
8  plot ( t, e, 'k.', 'markersize', 30 );
9  hold ( 'off' );

```



*Continuous and discrete models of proofreading*

## 2 A Model of Digestion and Excretion

The human digestive system uses a filtering process to gradually “subtract” material from the stomach and “add” it to the bloodstream. Similarly, material added to the bloodstream is gradually eliminated (by the kidneys, liver, and intestines) by a filtering process.

We will now try to model this process, in which medicine is received by the stomach, gradually filtered into the bloodstream, and then filtered out of the system entirely. Our goal is to prescribe a dosage amount and frequency for our patient, over a 48 hour treatment period, so that the medicinal level in the bloodstream reaches the therapeutic level, without becoming toxic.

To model the dosage problem, we want to represent the location and amount of the drug. We suppose that, at any time  $t$ , the medicine is divided up into three portions:

1.  $A$ : the medicine in the digestive system;
2.  $B$ : the medicine in the bloodstream;
3.  $C$ : the medicine that has left the body;

Now we need to make some assumptions about how these quantities change.

1. a dosage of  $\delta$  is added to  $A$  at regular intervals  $p$  times a day;
2. the quantity  $A$  is continuously filtered, at a rate  $\alpha$ , transferring medicine from the stomach  $A$  to the bloodstream  $B$ ;
3. the quantity  $B$  is continuously filtered, at a rate  $\beta$ , transferring medicine from the bloodstream  $B$  to the exterior  $C$ ;

To model how the medicine trickles through the system, we can write the following equations, where  $\delta(t)$ , the dosage supplied at time  $t$ , is zero except when the next scheduled dose is to be administered.

$$\begin{aligned} \frac{dA}{dt} &= +\delta(t) - \alpha * A(t) \\ \frac{dB}{dt} &= \quad \quad + \alpha * A(t) - \beta * B(t) \\ \frac{dC}{dt} &= \quad \quad \quad \quad + \beta * B(t) \end{aligned}$$

### 3 Programming the Model

The core of our MATLAB program uses the Euler method, discretizing the time span 0 to 48 hours into  $n + 1$  values, initializing  $A$ ,  $B$ , and  $C$  to zero, and taking  $n$  steps to advance the data from the current time to the next. Note that  $\delta$  on the right hand side of the ODE for  $A$  is discontinuous; it is zero except for moments when the next dosage is to be swallowed.

```

1  step_next_dose = 1;
2
3  for j = 1 : N
4
5      if ( j == step_next_dose )
6          delta = dosage;
7          step_next_dose = j + steps_between_doses;
8      else
9          delta = 0.0;
10     end
11
12     AtoB = dt * alpha * A(j);
13     BtoC = dt * beta * B(j);
14
15     t(j+1) = t(j) + dt;
16     A(j+1) = A(j) + delta - AtoB;
17     B(j+1) = B(j) + AtoB - BtoC;
18     C(j+1) = C(j) + BtoC;
19
20 end

```

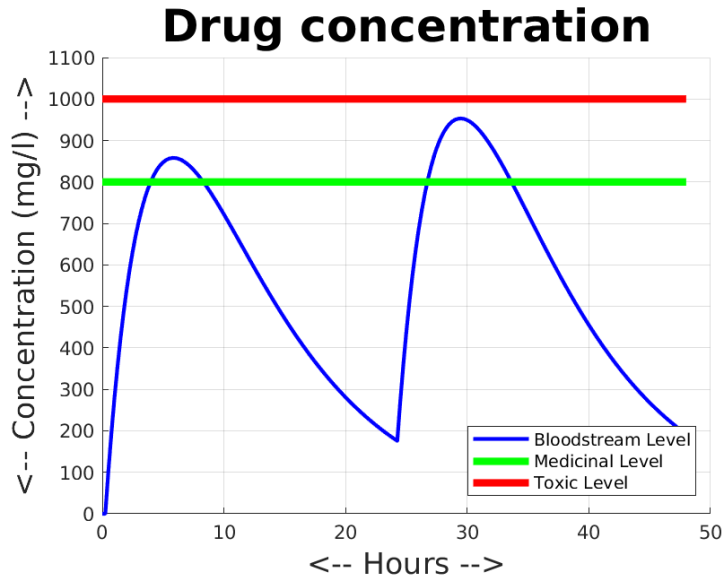
### 4 Numerical Experiments (No Humans Were Harmed...)

Now that we have a model, we make additional assumptions. The medicine is assumed to be beneficial only when the bloodstream level is at least 800 mg/l. To effect a cure, we need to exceed this level in the bloodstream for at least 15 hours. However, the medicine actually becomes toxic (causing injury or death) if its level ever exceeds 1000 mg/l. A total daily dose of the medicine is typically between 5,000 and 20,000 mg. We should compare this to the product  $p * \delta$ . The blood volume in our patient is 4.6 liters. The filter rates are  $\alpha = 0.25$  and  $\beta = 0.1155$ .

Now we seek a regimen of dosage amounts and times so that we can cure the disease, and not kill the patient. This means we have to experiment with

- $\delta$ , the amount, in milligrams, of a single dose;
- $p$ , the number of doses in a day;

Here, we try a scheme in which  $\delta = 7500$  and  $p = 1$ :



$$\delta = 7500 \text{ and } p = 1$$

From the plot, it's clear that we haven't killed the patient, but we also haven't kept the medicinal level about the 800 mg level for the required 15 hours. Now, using MATLAB, let's try to experiment with altering the values of *dosage* and *p* so that our patient gets better!