4452 Mathematical Modeling Lecture 12: Discrete and Continuous Simulation

Introduction

So far, the models we have examined and the systems which they model have been *deterministic*. This means that the outcome of the models depends on the initial conditions of the problem. The behaviour of the system for the same initial conditions will always be the same. Deterministic models typically involve the sensitivity analysis of the solution to various parameters in the model, since the solution is the same if the parameters don't change.

The word *simulation* is sometimes used as synonymous with *modeling*. However, we shall use it to describe modeling when the system being modeled is very complex, or you really don't have enough information to model the system as a deterministic system, or there are small random effects in the model. The solution depends on these random effects, or parameters which you cannot define exactly, and so you are forced to set up a model and solve it for a variety of different values of the unknown parameters, and see how the solutions vary. Simulations therefore typically involve solving the model for a variety of different values for parameters in the system, and then making predictions based on what you find. You would never base a prediction on a single run of a simulated system.

> **Example** British Gas transport natural gas under high pressure around Britain prior to its local distribution. When North Sea natural gas was first available, they had the problem of guaranteeing to customers, both industrial and domestic, that sufficient gas would always be available. There was no available pipe network which could be used to transmit the gas. Another complication is that the demand for gas fluctuates with seasonal and daily weather conditions.
>
> A mathematical model was formulated, based upon the known physical properties of gas and how it would flow along long large, very long pipes. The model was then operated under various conditions to simulate the real system. A number of different pipe layouts were tried to obtain an optimal design. Again, only after many simulations had led to a confident prediction that gas demand could be met in all circumstances was the construction of the pipe network authorised. [1]

In the above example the flow of gas through a given pipe system would be modeled by differential equations, and hence deterministic. However, the actual layout of the pipes and fluctuations of demand would be modeled by simulation.

Our textbook may leave you with the impression that simulation means using a numerical method to solve a differential equation. I think there is far more to it than that, as the above example illustrates.

Later on, we will look at *stochastic* models, where random effects play a central role in the model. A stochastic model takes on different values each time it is observed. Some authors use the term simulation whenever they use random numbers in a model.

Comparison of Discrete and Continuous Dynamical System Models (The Final Word)

We have already spent a bit of time looking at continuous and discrete models and identifying how they are different, and how they are the same.

If our continuous (differential equations) model is very complicated, it is likely that no analytic solution is available to us. We would prefer to have an analytic solution, since that may make our simulations easier to analyse. But that is not always possible, and we usually have to turn to a numerical method to solve a system of differential equations.

In my research (in theoretical physics and chemistry), I work with molecular collisions, and I calculate the path that the molecules take by solving the system of differential equations which describe a classical trajectory on a potential energy surface. For a typical problem, you may need to solve for 100,000 or more trajectories! Obviously, I would dearly love to have an analytical solution for the trajectory for a variety of initial conditions, but for the potential energy functions of real molecules that is not possible (at this time). I am forced to use a computer to solve these trajectories, and I solve them using Runge Kutta Order 4 (see Assignment 4).

What this means is that I am modeling reality (molecular collision) by a continuous model (differential equations, potential energy surface), and then that continuous model is solved on a computer, which is essentially a discrete model! My favourite line in the text addresses this point: "We certainly can't expect the computer to calculate $x(t)$ for every value of t. That would take an infinite amount of time to get nowhere." [3]

Although Runge Kutta is a more powerful and accurate differential equation solver, the comparison of discrete and continuous are best carried out in terms of Euler's method, since the comparison is very easy to understand.

Consider the discrete dynamical system, given by the following difference equation:

$$\begin{aligned}
x_1(n) - x_1(n-1) = f_1(x_1(n-1), x_2(n-1)) &= x_2^2(n-1) - 1 - x_1(n-1) \\
x_2(n) - x_2(n-1) = f_2(x_1(n-1), x_2(n-1)) &= x_1^2(n-1) - 1 - x_2(n-1), \\
x_1(0) = \alpha_1 = 1/2, & \\
x_2(0) = \alpha_2 = -1, & \\
n = 1, 2, 3, \dots, &
\end{aligned} \tag{1}$$

we iterate and find the sequence in *Mathematica* as follows (the solution is plotted in Fig. 1):

```
f1[x_, y_] = y^2 - 1 - x;
f2[x_, y_] = x^2 - 1 - y;
Clear[x1, x2]
x1[0] := a1
x2[0] := a2
x1[n_] := x1[n] = f1[x1[n - 1], x2[n - 1]] + x1[n - 1]
x2[n_] := x2[n] = f2[x1[n - 1], x2[n - 1]] + x2[n - 1]
num = 10;
data = Table[{n, x1[n], x2[n]}, {n, 0, num}];
data = data /. a1 -> 0.5 /. a2 -> -1.0
phase = Table[{Part[data, n + 1, 2], Part[data, n + 1, 3]}, {n, 0, num}];
datax1 = Table[{Part[data, n + 1, 1], Part[data, n + 1, 2]}, {n, 0, num}];
datax2 = Table[{Part[data, n + 1, 1], Part[data, n + 1, 3]}, {n, 0, num}];
ListPlot[phase, PlotJoined -> True, Frame -> True,  FrameLabel -> {"x1", "x2"}]
ListPlot[datax1, PlotJoined -> True, Frame -> True, FrameLabel -> {"n", "x1"}]
ListPlot[datax2, PlotJoined -> True, Frame -> True, FrameLabel -> {"n", "x2"}]
```

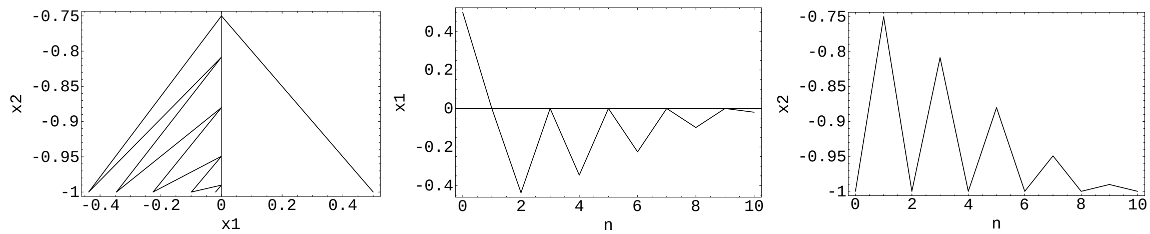

Figure 1: The phase space solution to the discrete system in Eq. (1).

Now, let's consider the continuous analogue of the discrete system in Eq. (1). It would be

$$\begin{aligned}\frac{dx_1}{dt} = f_1(x_1, x_2) &= x_2^2 - 1 - x_1 \\ \frac{dx_2}{dt} = f_2(x_1, x_2) &= x_1^2 - 1 - x_2, \\ & x_1(0) = \alpha_1 = 1/2, x_2(0) = \alpha_2 = -1, \end{aligned} \tag{2}$$

To solve this on a computer, we can approximate the derivatives by

$$\begin{aligned}\frac{dx_1}{dt} &\sim \frac{x_1(t_i) - x_1(t_{i-1})}{t_i - t_{i-1}} \\ \frac{dx_2}{dt} &\sim \frac{x_2(t_i) - x_2(t_{i-1})}{t_i - t_{i-1}}\end{aligned}$$

and if we identify $h = t_i - t_{i-1}$ as the time step, we have

$$\begin{aligned} x_1(t_i) = h\, f_1(x_1(t_{i-1}), x_2(t_{i-1})) + x_1(t_{i-1}) &= h\,(x_2^2(t_{i-1}) - 1 - x_1(t_{i-1})) + x_1(t_{i-1}) \\ x_2(t_i) = h\, f_2(x_1(t_{i-1}), x_2(t_{i-1})) + x_2(t_{i-1}) &= h\,(x_1^2(t_{i-1}) - 1 - x_2(t_{i-1})) + x_2(t_{i-1}), \\ & x_1(0) = \alpha_1 = 1/2, x_2(0) = \alpha_2 = -1, \\ & i = 1, 2, 3, \ldots \end{aligned} \tag{3}$$

Our continuous system in Eq. (2) is approximated by the system in Eq. (3). They will become the same in the limit as $h \to 0$. Computationally, you do not want to take $h = 0$ (*That would take an infinite amount of time to get nowhere*), but you do want it to be small. How small is small enough? You will have to do some testing to find out. If $h = 1$, our approximation is identical to the discrete system in Eq. (1).

In *Mathematica*, this can be solved in the following manner (notice I've removed the a1 and a2 initial assignments for $x_1(0)$ and $x_2(0)$ to speed things up):

```
Clear[x1, x2]
x1[0] := 0.5;
x2[0] := -1.0;
num = 10;
to = 0;
tf = 10;
```

```
h = N[(tf - to)/num]
x1[n_] := x1[n] = h f1[x1[n - 1], x2[n - 1]] + x1[n - 1]
x2[n_] := x2[n] = h f2[x1[n - 1], x2[n - 1]] + x2[n - 1]
data = Table[{h*n, x1[n], x2[n]}, {n, 0, num}];
phase = Table[{Part[data, n + 1, 2], Part[data, n + 1, 3]}, {n, 0, num}];
datax1 = Table[{Part[data, n + 1, 1], Part[data, n + 1, 2]}, {n, 0, num}];
datax2 = Table[{Part[data, n + 1, 1], Part[data, n + 1, 3]}, {n, 0, num}];
ListPlot[phase, PlotJoined -> True, Frame -> True,  FrameLabel -> {"x1", "x2"}]
ListPlot[datax1, PlotJoined -> True, Frame -> True, FrameLabel -> {"n", "x1"}]
ListPlot[datax2, PlotJoined -> True, Frame -> True, FrameLabel -> {"n", "x2"}]
```

With num (the number of points) equal to ten we get the discrete case ($h = 1$). By increasing the value of num, we approach the continuous case. The results are plotted in Fig. 2.

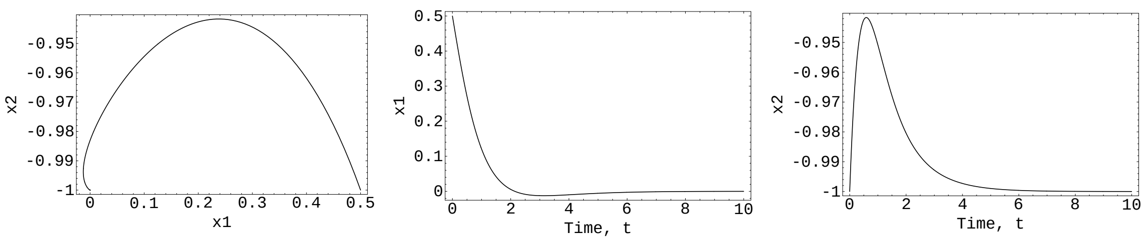

Figure 2: The phase space solution to the continuous system in Eq. (3). The value of num is 500, which means we have a step size of $h = 0.02$.

A plot of the continuous and discrete models is overlaid on the vector field for this system in Fig. 3. The behaviour is quite different, due to the time delay in the action of the system for the discrete case. One is not more correct than the other; the one you should choose depends on the behaviour of the system you are modeling.

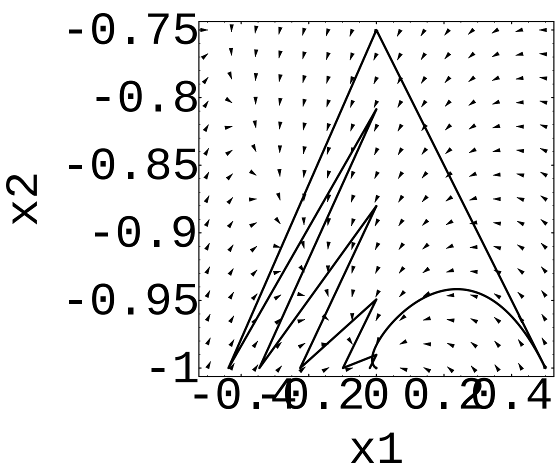

Figure 3: The phase space solution to the continuous system in Eq. (3) and discrete system Eq. (1). The vector field is the same for both cases.

Simulation

As mentioned earlier, simulation can mean different things to different people. I tend to think of it as what you do when you are not told enough information to completely model a problem, and you have to assign parameters in the problem values before you can solve it (you as the model creator do this, the numbers are not given to you). Since you are not sure if the numbers are "correct", you should do a number of solutions of the model for varying values of the parameters you have chosen. There is one thing anyone who uses the word simulation agrees upon–a simulation *requires* more than one solution of the model be computed.

The reasons why a person would use a simulation model can be quite varied. They could include collecting statistics on the long-term behaviour of a system, comparing alternate arrangements of a system, investigating the effects of changing the parameters, investigating the effects of altering the modeling assumptions, and finding optimal operating conditions for the system. [2]

Problem 6.5.1 Revisiting the battle simulation of Example 6.1.

We are interested in examining how weather effects the outcome of the battle. Bad weather and poor visibility decrease the effectiveness of direct fire weapons for both sides. Indirect fire weapons are not affected by weather conditions. The effect of bad weather can therefore be introduced by introducing a parameter w in our model, $w = 1$ represents good weather (visibility), and $w = 0$ represents horrible weather (visibility). How does the result of the battle change under changed weather conditions?

The new equations which incorporate the effects of weather are:

$$\begin{aligned} \Delta x_1 &= -w\lambda(0.05)x_2 - \lambda(0.005)x_1x_2 \\ \Delta x_2 &= -w(0.05)x_1 - (0.005)x_1x_2 \end{aligned} \tag{4}$$

Note: In this example we are assuming the weather remains the same throughout the battle, so w is a constant. A *stochastic* model would assume that the weather could vary day to day, and hence w would not be a constant, it would be a random variable that changed each day (but not completely random).

Using the following *Mathematica* code, I ran a number of simulations, and the results are presented in Fig. 4 and Table 1.

```
Do[{Clear[x1, x2, f1, f2],
    f1[x_, y_] = -w Lambda(0.05) y - Lambda(0.005) x y,
    f2[x_, y_] = -w 0.05x - 0.005x y,
    x1[0] := 5,
    x2[0] := 2,
    num = 70,
    x1[n_] := x1[n] =  f1[x1[n - 1], x2[n - 1]] + x1[n - 1],
    x2[n_] := x2[n] =  f2[x1[n - 1], x2[n - 1]] + x2[n - 1],
    data = Table[{n, x1[n], x2[n]}, {n, 0, num}],
    Do[{If[ Part[data, n + 1, 2] < 0 || Part[data, n + 1, 3] < 0,
            {Print["Battle ends after ",Part[data, n + 1, 1], " hours. w = ", w,
              " lambda = ", Lambda], Break[]}]}, {n, 0, num}],
    phase = Table[{Part[data, n + 1, 2], Part[data, n + 1, 3]}, {n, 0, num}],
    contphase =
      ListPlot[phase, PlotRange -> {{0, 5}, {0, 2}}, PlotJoined -> True,
        Frame -> True]},
{Lambda, 1, 5}, {w, 0.2, 1, 0.2}]
```

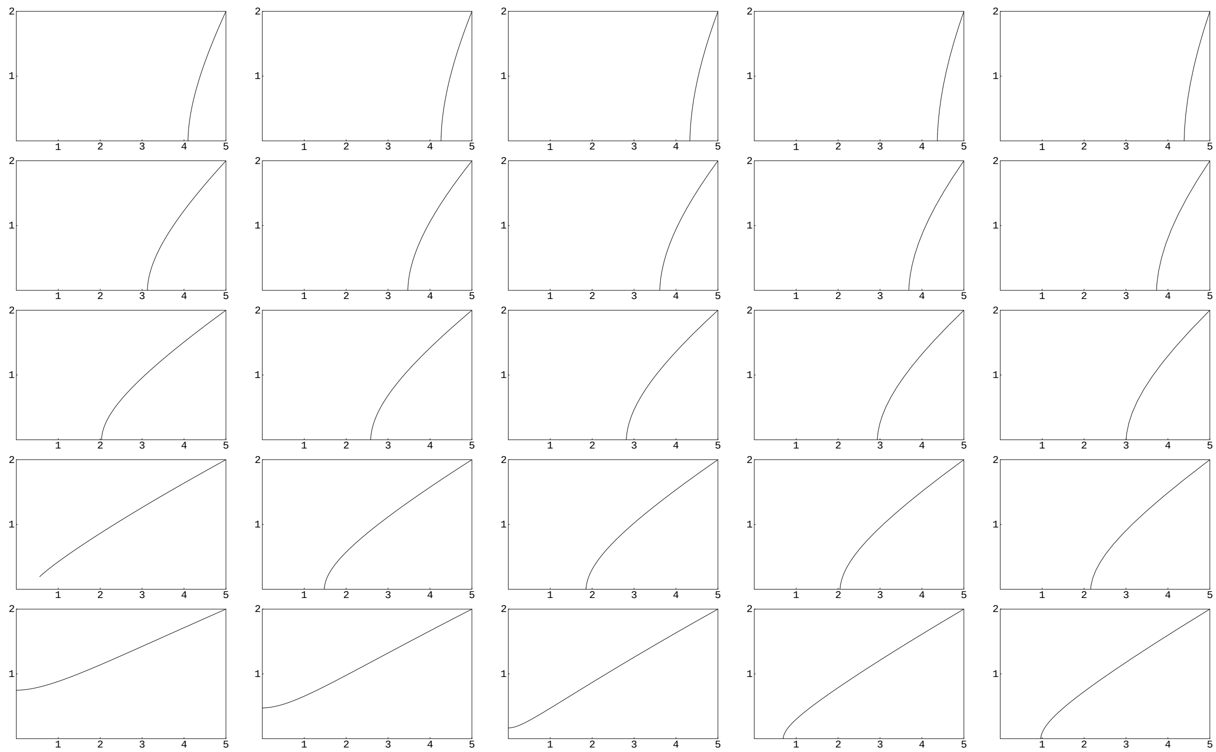

Figure 4: The results of simulating a battle between two forces of unequal size under varying weather conditions ($0 < w < 1$) and greater weapon effectiveness of the smaller force ($\lambda > 1$). The model is Eq. (4). Initially, there are 5 Red Divisions (Red is along the horizontal axis) and 2 Blue Divisions (vertical). The axis labels have been omitted for clarity. The columns correspond to $w = 0.2, 0.4, 0.6, 0.8, 1.0$; the rows $\lambda = 1, 2, 3, 4, 5$.

λ	w	Duration	λ	w	Duration	λ	w	Duration	λ	w	Duration	λ	w	Duration
1	0.2	32	1	0.4	18	1	0.6	13	1	0.8	10	1	1.0	8
2	0.2	38	2	0.4	21	2	0.6	14	2	0.8	11	2	1.0	9
3	0.2	49	3	0.4	25	3	0.6	17	3	0.8	13	3	1.0	10
4	0.2	–	4	0.4	33	4	0.6	21	4	0.8	15	4	1.0	12
5	0.2	46	5	0.4	37	5	0.6	41	5	0.8	25	5	1.0	17

Table 1: The duration of the battle modeled by Eq. (4). There was one battle simulation ($\lambda = 4, w = 0.2$) which did not result in a victory for either side after 70 hours.

Our simulation found that the Red won 21 of the battles, Blue won 3, and one battle had not resulted in a win for either Blue or Red after 70 hours. It appears from our simulation that for Blue to win they require significantly better weapons than the larger Red forces; at least 5:1 weapons superiority. Even with a 5:1 weapons superiority, Blue will not win the battle unless they receive help from bad weather. Under poor weather conditions the performance of the Blue forces improved. Also, the battles take longer to complete under poor weather conditions.

If we required more information, like the number of divisions left when the battle is over, we could get that information by outputting it at the same time as the battle time information. I also modeled this as a continuous problem, and the results did not change significantly.

References

[1] D. Edwards & M. Hamson, Guide to Mathematical Modelling, CRC Press (Baco Raton) 1990 (p42).

[2] D. Edwards & M. Hamson, Guide to Mathematical Modelling, CRC Press (Baco Raton) 1990 (p144).

[3] M. Meerschaert, Mathematical Modelling, 2nd ed., Academic Press (San Diego) 1999 (p200).