

The Mortgage Payment Problem: Approximating a Discrete Process with a Differential Equation

Michael Monagan.

Department of Mathematics,
Simon Fraser University,
Vancouver, British Columbia, V5A 1S6
Canada
mmonagan@sfu.ca

The differential equation

$$y'(t) = k \cdot y(t) \text{ with } k > 0$$

models exponential growth. Here the constant k is the natural population growth rate. We use it to model biological populations (people, fish, bacteria) which are discrete processes. How good is the approximation? Well that depends. In an animal population, e.g. a duck or sheep population, births occur in spring which means births are neither continuous nor uniform. And any number of factors could result in actual population numbers not exhibiting exponential growth. Logistic growth is another population growth model that is often introduced early in a course covering differential equations as a model with limited growth. The logistic equation is

$$y'(t) = a \cdot y(t) \cdot (C - y(t)) \text{ where } a > 0 \text{ and } C > 0.$$

The constant C is called the *carrying capacity* of the population and the constant a is related to the natural population growth rate by $C \cdot a = k$. But how accurate is this model? At Simon Fraser University, we use the Stewart Calculus text for first year calculus. Chapter 9 includes a section focusing on the logistic model. No attempt is made, however, to examine how good the model is in approximating any limited growth process. When we teach differential equations we really don't spend much time, if any, determining how well differential equations approximate real data or discussing limitations of the models. It's impractical to ask students to measure population data. And if we give them some data, if they try to fit the logistic model to the data they will end up with a non-linear least squares fitting problem which is too difficult to do in the first year. Another difficulty is that any attempt to do this will not lead to nice simple questions that we can ask students on a final exam! What would be helpful is an example of a process which we can simulate readily and which can be modeled accurately by a DE so that the student may investigate how accurately the DE approximates the process. The example presented here is such an example. It presumes that the student has access to a mathematical software package like Maple and that the student can write simple programs with loops.

Paying down a Mortgage

Suppose you take out a mortgage for **\$200,000** to buy a house. Suppose the mortgage is for **30 years** and suppose the interest rate charged by the bank is $r = 5\%$ per year compounded daily, that is you pay $r/365\%$ per day. What is the payment **\$P** per year that you need to make so that after 30 years you have paid off the mortgage? And how much interest will you pay in total over the lifetime of the mortgage? That's an interesting question that the bank will probably not answer. They will trot out some answer like, "well, the interest rate will change in 30 years so we don't know how much you will actually pay." But surely they can tell you how much you will pay assuming the interest rate doesn't change. Maybe the bank doesn't want you to know so that you don't try to pay off the mortgage as fast as possible!

So how does one pay down a mortgage? Usually one makes periodic payments, either monthly or biweekly. That is one pays $\$P/12$ per month at the end of the month, or, approximately $\$P/26$ every second week. Thus the daily interest charges and periodic payments mean the process is discrete. Yet over 30 years the amount owed at time t will look like a smooth function. So we can approximate the process using a differential equation. Let's set it up.

Let $M(t)$ be the amount owed on a \$200,000 mortgage at time t years.

Suppose the annual interest rate on the mortgage is $r = 5\%$.

Suppose the term of the mortgage is 30 years, i.e., $M(30)$ should be 0.

Suppose we pay $\$P$ per year. We can use the following differential equation to model interest charges and our payments.

$$M'(t) = r \cdot M(t) - P \text{ dollars per year.}$$

Now let's find the value of P such that $M(30) = 0$. This example illustrates why getting a formula for the answer from Maple is helpful.

```
> restart;
de := diff( M(t), t) = r*M(t) - P;
r := 0.05;
```

$$de := \frac{d}{dt} M(t) = r M(t) - P$$

$$r := 0.05 \tag{1}$$

We solve the DE in Maple with initial value $M(0) = \$200,000$.

```
> sol := dsolve( {de, M(0)=200000}, M(t) );
```

$$sol := M(t) = 20 P + e^{\frac{1}{20} t} (200000 - 20 P) \tag{2}$$

To determine P we solve $M(30) = 0$ for P .

```
> eq := eval( rhs(sol), t=30.0 ) = 0;
```

$$eq := -69.63378140 P + 8.963378140 10^5 = 0 \tag{3}$$

```
> yearpayment := solve( eq, P );
```

$$yearpayment := 12872.16917 \tag{4}$$

So we pay \$12,872.17 dollars per year.

So if the bank requires us to make equal monthly payments (at the end of each month), we pay

```
> monthpayment := yearpayment/12;
```

$$monthpayment := 1072.680764 \tag{5}$$

per month. And the total interest we pay is

```
> interestpaid := 30*yearpayment - 200000;
```

$$\$186,165.07 \tag{6}$$

Notice that we pay almost as much interest as the value of the mortgage!

How accurate is the estimate for P?

What I do now is show the students how to simulate daily interest charges and daily mortgage payments then ask them to see how accurate the model is using monthly payments. We start with $M = 200,000$ and simulate the daily interest charges and payments over the 30 years using $P =$

$\$11448.10$ per year by adding $\frac{r}{365} \cdot M$ to M daily and then subtracting $\frac{P}{365}$ from M .

```
> M := 200000;
for y to 30 do
```

```

    for d to 365 do
      M := M + r/365*M;
      M := M - yearlypayment/365;
    od;
od:

```

\$200,000.00 (7)

```

> M;

```

\$26.78 (8)

If the model is accurate we should end up with M close to 0 and we do. Now for monthly payments, since months have different lengths, I ask the students to figure out on which day of the year each month ends. For simplicity, I tell them to assume there are no leap years so February always has 28 days. Here is my solution. Note, these simulations mean that the students need to know how to write simple programs. Most students, even though they have taken a first programming course have some difficulty with simple programs like these.

```

> DaysOfMonth := [31,28,31,30,31,30,31,31,30,31,30,31];
    DaysOfMonth := [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

```

(9)

```

> M := 200000;
  for y to 30 do
    for m to 12 do
      for d to DaysOfMonth[m] do M := M + r/365*M; od;
      M := M - monthlypayment;
    od;
od:

```

200000 (10)

Now, as expected, monthly payments is further from 0 than daily payments.

```

> M;

```

1733.447541 (11)

Note, my Maple solution is deceptively simple. One of the surprises to me was the large variety of different ways the students figured out how to encode that simulation. Several coded it the following way. They first typed in on which day of the year the end of each month occurred. Some got these numbers wrong because they cannot add. Writing a loop to compute these numbers would have been wise.

```

> endofmonth := [31,59,90,120,151,181,212,243,273,304,334,365];
    endofmonth := [31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365]

```

(12)

```

> M := 200000;
  for y to 30 do
    for d to 365 do
      M := M + r/365*M;
      if member(d,endofmonth) then
        M := M - monthlypayment;
      fi;
    od;
od:

```

200000 (13)

```

> M;

```

1733.447541 (14)

Now the result shows that the simulation is not that accurate if we use $P = \$11448.10$. We still owe $M = \$1,733.44$. Why? Is it because the value of P is too small? No, it's because the simulation pays down the mortgage monthly **at the end of the month**. What I ask the students to do is to repeat this calculation assuming that we make the payments at the middle of each month. We should find that M is

closer to 0 after 30 years. To see this accurately we should see what happens if we make payments for each day of the month and see which results in M closest to 0. Hence

```

> for day from 12 to 19 do
  M := 200000;
  for y from 1 to 30 do
    for m from 1 to 12 do
      for d from 1 to DaysOfMonth[m] do
        M := M + r/365*M;
        if d = day then
          M := M - monthlypayment;
        fi;
      od;
      if day > DaysOfMonth[m] then
        M := M - monthlypayment;
      fi;
    od;
  od;
  result[day] := M;
od;
> seq( day=result[day], day=12..19 );
12 = -525.5457571, 13 = -402.7229930, 14 = -279.9019422, 15 = -157.1221971, 16 =
-34.33205206, 17 = 88.43804809, 18 = 211.1759193, 19 = 333.9008988

```

Thus paying on the 16th day of the month results in M being closest to 0 after 30 years. And it is very close at M = -\$34.44, almost as close as daily payments.

Visualizing the discrete payment process.

To see the payment process more clearly, let us graph the process for the first 100 days assuming biweekly payments this time. Asking the student to assemble graphs of functions and data is an important skill that they should acquire. Maple is helpful here because you can create and assign plots to programming variables (P1, P2 and P3 below) and then display them together using the plots [display] command. This is an extremely useful feature that allows us to create any plot we want.

```

> 365-26*14;
1

```

```

> sol := eval( rhs(sol), P=yearlypayment );
sol := 2.574433834 105 - 57443.3834 e $\frac{1}{20}t$ 

```

```

> sol;
2.574433834 105 - 57443.3834 e $\frac{1}{20}t$ 

```

```

> numdays := 100;
100

```

P1 is a plot of $M(t)$ \$thousand for 100 days.

```

> P1 := plot( sol/1000, t=0..numdays/365 );
P1 := PLOT(...)

```

Note, in this my second version, I have scaled the \$ figures to thousands for display.

```

> biweeklypayment := yearlypayment/365*14.0;
493.7270366

```

```

> M := 200000;
Data := Array(0..2*numdays):

```

```

Data[0] := [0,M/1000]:
for d to numdays do
  Data[2*d-1] := [d/365,M/1000]; # before interest
  M := M + r/365*M;
  if d mod 14 = 0 then M := M - biweeklypayment fi;
  Data[2*d] := [d/365,M/1000];
od:

```

P2 here is a plot of a sequence of $[x, y]$ points joined together by straight lines.

```
> P2 := plot( Data, style=line );
```

P3 is a plot of some text at co-ordinates (0.01,198.5).

```
> P3 := plots[textplot](
  [0.01,198.5,"Plot of $thousands owed verses time (years) for 100
  day simulation"],
  align=right, font=[HELVETICA,14] );
```

```
      P3 := PLOT(...)
```

(22)

```
> plots[display]( [P1,P2,P3], view=[default,198..201] );
```

