

Math Tools: Recursive Methods

Revised: November 28, 2015

The concept of recursion runs throughout modern mathematics and computer science. The same is true of economics and finance. There's a reason the leading PhD textbook in macroeconomics is called *Recursive Macroeconomic Theory*. Such work in economics reflects, in large part, the adoption by economists of methods developed elsewhere.

What follows is a short informal introduction to the idea and a start on the kinds of applications you'll find in economics and finance.

Warning: This contains a little linear algebra. Skip it if that's not part of your skill set.

1 Examples of recursion

The idea is to characterize a sequence of items, indexed by an integer $n = 0, 1, 2, \dots$, by a rule that connects each item to the next one. If we label the items x_n , the rule might be expressed

$$x_{n+1} = g(x_n). \quad (1)$$

If we have a starting point, say x_0 , the rule tells us how to compute as many succeeding items as we wish. We would say that the set $\{x_n\}$ is generated recursively and refer to (1) as the defining recurrence relation. A “solution” to (1) is a formula that expresses x_n as a function of n .

Examples:

1. *Linear difference equation.* Let

$$x_{n+1} = ax_n. \quad (2)$$

This has the solution $x_n = a^n x_0$. It converges to zero if $|a| < 1$, but it's the solution either way.

2. *Logistic map.* Let

$$x_{n+1} = ax_n(1 - x_n)$$

with $0 < a \leq 4$. If you try some experiments, you'll see that it generates wildly different behavior depending on the value of a . You might set $x_0 = 0.3$ and $a = (0.98, 1.5, 2.5, 3.25, 3.5)$, generate (say) 20 terms, and graph the output. See [Wikipedia](#). The point, which we won't develop further, is that even quite simple nonlinear recurrences can generate complex behavior.

3. *Fibonacci numbers.* The Fibonacci numbers are generated by the second order system

$$f_{n+1} = f_n + f_{n-1}$$

starting with $f_0 = 0$ and $f_1 = 1$. In matrix terms, we can write this as $x_{n+1} = Ax_n$ with

$$x_n = \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

This is similar to (2), but here x_n is a vector. The matrix A has eigenvalues (λ_1, λ_2) satisfying $\lambda^2 - \lambda - 1 = 0$. The “solution” has the form $f_n = c_1\lambda_1^n + c_2\lambda_2^n$ for constants (c_1, c_2) that satisfy the initial conditions.

This is a common example in computer science courses. A recursive version of a Matlab program to compute Fibonacci numbers is

```
function answer = f(n)
if n==0
    answer = 0;
elseif n==1
    answer = 1;
else
    answer = f(n-1) + f(n-2);
end
end
```

Note that the function `f` refers to itself — it’s [recursive](#) in the sense the word is used in computer science. In Matlab, we would save this as a file called `f.m`, then call it by typing (say) `f(8)` in the command line or as a line in another program. (If you enter a fraction, it blows up, so a better function would check and generate an error message, or perhaps convert `n` to an integer.)

4. *Mean and variance.* [John Cook](#) describes the Welford method of computing the mean and variance recursively. Consider a sequence of observations: x_1, x_2, \dots . We can compute rolling estimates of the mean and variance from

$$\begin{aligned} M_n &= M_{n-1} + (x_n - M_{n-1})/n \\ S_n &= S_{n-1} + (x_n - M_{n-1}) * (x_n - M_n) \end{aligned}$$

starting with $M_1 = x_1$ and $S_1 = 0$. Do a few terms to assure yourself that M_n is the mean of the first n observations and S_n is the sum of the squared deviations from the mean. The standard estimator of the variance s^2 is therefore $S_n/(n-1)$ (although I prefer to divide by n , always).

5. *Natural numbers.* The natural numbers are the set $\mathbb{N} = \{0, 1, 2, \dots\}$. We can define them recursively with the rules: (i) 0 is in \mathbb{N} and (ii) if n is in \mathbb{N} then so is $n+1$.

Which reminds me of an old George Gamov story. In the story, the Hilbert Hotel has an infinite number of rooms numbered $1, 2, 3, \dots$. By law, it must save one for the King, but the innkeeper fills them all anyway. When asked, he says: “No problem, I can always get

an open room by asking everyone to move over one.” And if we move the person in room j to room $2j$, we open up an infinite (countable) number of rooms, all the odd-numbered ones.

6. *Combinatorics.* Computer scientist [Herbert Wilf](#) notes that many combinatoric identities satisfy recurrences. For example, the binomial coefficients,

$$f(n, k) = \binom{n}{k} = \frac{n!}{k!(n-k)!},$$

are the solution to

$$f(n, k) = f(n-1, k) + f(n-1, k-1)$$

starting with $f(n, 0) = 1$.

7. *Functions.* In economics and finance we often run across recursion with functions. Suppose we have a sequence of functions $f_n(x)$ over some domain x that satisfy the recurrence

$$f_{n+1}(x) = g[f_n(x)]$$

for some g . Even better, suppose the recurrence has a fixed point:

$$f(x) = g[f(x)].$$

Here we have an equation in which the unknown is another function f rather than a number x . It’s similar to (1), but we’re dealing with a more complex object.

2 Recursion in asset pricing

Similar methods show up throughout economics and finance. The idea is to string together a series of one-period steps — recurrences — similar to those we used earlier in the course. That allows us to approach the price of (say) an n -period bond with the same methods we used to price a one-period bond.

We’ll do all this in Markov settings, which require some notation. You’ll recall that modern asset pricing is based on the no-arbitrage theorem: there exists a positive pricing kernel m that satisfies $E(mr) = 1$ for returns r on all assets. In a Markov environment, we need to keep track of the current state z_t and the possible future states z_{t+1} . The ingredients include:

- Probabilities. We have a state variable z_t and conditional probabilities $p(z_{t+1}|z_t)$.
- Returns. One-period returns from date t to $t+1$ depend on the state in both periods: $r(z_t, z_{t+1})$.
- Asset pricing. The no-arbitrage theorem becomes: there exists a positive $m(z_t, z_{t+1})$ satisfying

$$E_t[m(z_t, z_{t+1})r(z_t, z_{t+1})] = 1 \tag{3}$$

for all returns $r(z_t, z_{t+1})$. Here E_t is the expectation conditional on the current state z_t — the expectation computed from $p(z_{t+1}|z_t)$, in other words.

Examples:

1. *Bond pricing.* A bond of maturity n is a claim to a payment of one in n periods. In a Markov setting, such bond prices are functions of the state. The question is what the functions are. We find bond prices recursively, starting with $q^0(z_t) = 1$ for all states z_t (a dollar today is worth a dollar). Bonds of longer maturity have returns

$$r^{n+1}(z_t, z_{t+1}) = q^n(z_{t+1})/q^{n+1}(z_t).$$

The no-arb theorem (3) then gives us the recursion

$$q^{n+1}(z_t) = E_t[m(z_t, z_{t+1})q^n(z_{t+1})]. \quad (4)$$

In words: an $n + 1$ -period bond is a claim to an n -period bond in one period.

We'll spend some time with a loglinear functional form. This takes some work, but it's worth doing because we'll be spending some time with similar models. Despite how it might look at first, this is a user-friendly functional form. Suppose the pricing kernel is loglinear:

$$\begin{aligned} \log m(z_t, z_{t+1}) &= \delta + az_t + bz_{t+1} \\ z_{t+1} &= \varphi z_t + \sigma w_{t+1} \end{aligned}$$

with $\{w_t\}$ a sequence of independent standard normal random variables and $0 < \varphi < 1$. Then bond prices are loglinear functions of the state:

$$\log q^n(z_t) = A_n + B_n z_t \quad (5)$$

for coefficients (A_n, B_n) to be determined.

The solution follows from applying (4) to (5). We start with

$$\log q^{n+1}(z_t) = \log E_t \{ \exp [\log m(z_t, z_{t+1}) + \log q^n(z_{t+1})] \}.$$

We get the left side from (5). The right side takes some work. The inside of the square brackets on the right can be expressed

$$\begin{aligned} \log m(z_t, z_{t+1}) + \log q^n(z_{t+1}) &= (\delta + az_t + bz_{t+1}) + (A_n + B_n z_{t+1}) \\ &= \delta + A_n + az_t + (b + B_n)(\varphi z_t + \sigma w_{t+1}). \end{aligned}$$

Conditional on the state z_t , this is normal with mean and variance

$$\begin{aligned} E_t [\log m(z_t, z_{t+1}) + \log q^n(z_{t+1})] &= \delta + A_n + [a + (b + B_n)\varphi]z_t \\ \text{Var}_t [\log m(z_t, z_{t+1}) + \log q^n(z_{t+1})] &= (b + B_n)^2 \sigma^2. \end{aligned}$$

The usual “mean plus variance over two” gives us

$$\begin{aligned} \log E_t (m(z_t, z_{t+1})q^{n+1}(z_{t+1})) &= \log E_t \{ \exp [\log m(z_t, z_{t+1}) + \log q^{n+1}(z_{t+1})] \} \\ &= \delta + A_n + [a + (b + B_n)\varphi]z_t + (b + B_n)^2 \sigma^2 / 2. \end{aligned}$$

[If this isn't clear, go through it again, it's important.] By assumption, this equals $A_{n+1} + B_{n+1}z_t$ for all values of z_t , so we must have

$$\begin{aligned} A_{n+1} &= \delta + A_n + (b + B_n)^2 \sigma^2 / 2 \\ B_{n+1} &= a + (b + B_n) \varphi. \end{aligned}$$

Evidently we've converted the recursion in $q^n(z_t)$, equation (4), into recursions in the coefficients (A_n, B_n) . They're not pretty, but we can easily compute them. The initial conditions $A_0 = B_0 = 0$ correspond to $\log q^0(z_t) = \log(1) = 0$.

2. *Equity pricing.* A dividend paying stock is a more complicated object. In the same environment as before, let the dividend in state z_t be $d(z_t)$. The ex-dividend value of a share might be expressed recursively as

$$v(z_t) = E_t \{ m(z_t, z_{t+1}) [d(z_{t+1}) + v(z_{t+1})] \}. \quad (6)$$

In words: equity today is a claim to two things tomorrow, a dividend and the same share of equity.

Note that the unknown in this equation is the function v . It's also recursive: you need to know v on the right to compute v on the left. You're now as ready as you'll ever be for the recursion joke: "To understand recursion, you need to understand recursion." Or Google "recursion." You get back: "Did you mean: recursion?"

One way to think about this is as the limit of a finite horizon. Suppose we value next period's dividend by

$$v^1(z_t) = E_t \{ m(z_t, z_{t+1}) d(z_{t+1}) \}.$$

The superscript 1 here means we're valuing one period of dividends. We can value two periods of dividends recursively with

$$v^2(z_t) = E_t \{ m(z_t, z_{t+1}) [d(z_{t+1}) + v^1(z_{t+1})] \}.$$

In general, we can value $n + 1$ periods of dividends with the recursion

$$v^{n+1}(z_t) = E_t \{ m(z_t, z_{t+1}) [d(z_{t+1}) + v^n(z_{t+1})] \},$$

starting with $v^0(z_t) = 0$ (the value of zero dividends is zero). As we increase n , we have more and more dividends. We might imagine, if all goes well, that as n gets larger and larger, we approach (6).

3. *Perpetual American options.* Consider the option to buy one share of stock next period for strike price k . The value today in state z_t is

$$q(z_t) = E_t \{ m(z_t, z_{t+1}) [v(z_{t+1}) - k]^+ \},$$

where $x^+ = \max\{0, x\}$. Evidently we exercise in states where $v(z_{t+1}) - k$ is positive and not in other states.

A perpetual option allows us to wait. If we don't exercise now, we can hold the option for another period. Valuation has a recursive form:

$$q(z_t) = \max \{ v(z_t) - k, E_t [m(z_t, z_{t+1}) q(z_{t+1})] \}.$$

That is: we either exercise now and get $v(z_t) - k$ (the first branch of the max) or continue to hold the option and get the current value of the option next period (the second branch).

3 Bottom line

Recursive methods are at the heart of modern macroeconomics and finance. We'll use them extensively to value bonds.

Practice problems

1. *Discounted cash flows.* Our goal here is to simplify the pricing relation (6) and derive a more conventional valuation of equity as the expected discounted value of future dividends.

- (a) Simplify (6) using $m(z_t, z_{t+1}) = \delta$ and replacing dependence on the state z_t with a subscript t — that is, by replacing $v(z_t)$ with v_t .
- (b) Use this simplification to express equity's value as

$$v_t = \sum_{j=1}^n \delta^j E_t(d_{t+j}) + \delta^n E_t(v_{t+n}).$$

- (c) What happens as n gets large? What happens to the second term on the right above?
- (d) What does this example leave out that's present in (6)?

Answer.

- (a) Equation (6) becomes

$$v_t = \delta E_t(d_{t+1} + v_{t+1}).$$

- (b) This follows from repeated substitution and the law of iterated expectations.
- (c) We hope that the second term goes to zero. That leaves us with the infinite sum

$$v_t = \sum_{j=1}^{\infty} \delta^j E_t(d_{t+j}).$$

- (d) If $m(z_t, z_{t+1}) = \delta$ there are no risk premiums. The price of equity depends only on expected future dividends.
2. *Consols.* A consol pays a constant coupon c every period — forever. They have been used by the British as a government financing tool since the 1700s. How would you adapt equation (6) to value such an instrument?

Answer. We set $d(z_t) = c$, giving us

$$v(z_t) = E_t\{m(z_t, z_{t+1})[c + v(z_{t+1})]\}.$$

The variation in price here comes from m , which translates roughly as variation in interest rates.