The mathematics of the “solera” system

Alasdair McAndrew
Alasdair.McAndrew@vu.edu.au
College of Sport, Health and Engineering
Victoria University
PO Box 14421, Melbourne, Victoria 8001
Australia

Abstract

Fractional blending, also known as the “solera system”, is a technique dating from the mid 19-th century, for the aging of liquids such as fortified wines, spirits, and balsamic vinegars. Such products require careful aging before they can be sold, and careful mixing of liquids from different ages is thus required. At each stage, every six months for example, or each year, a new un-aged liquid is added to the system, and a sequence of mixings is used to “filter”, as it were, this new material through the system. The result at the end is a liquid carefully blended from different ages, with the oldest predominating. When properly done, this ensures a constant supply of an appropriately aged product. The mathematics can be described as a sequence of difference equations, or recurrence relations, which leads into some matrix algebra, and it turns out that this mathematics is more interesting than the simple explanation of the system might lead one to believe. This article explores this mathematics, using a computer algebra package for all the heavy lifting.

1 Introduction

The *solera system* [7] originated in Spain in the 19th century. It consists of a selection of barrels—of sherry, for example—all of which together form the *criaderas*, or nursery.

A solera system is generally visualized as a pyramid, as shown in figure 1. The bottom barrels are in fact the solera, which means “floor” or “sole” in Spanish.

Although a pyramid is a standard representation of a solera system, in fact the different rows of barrels may be spread out in different cellars, just very carefully labelled.

At the end of each aging period; a year say, or maybe six months, one-quarter of the sherry in each of the bottom barrels is taken away for bottling. They are filled up from the third row; each barrel of which loses one-third of its contents. These barrels, in turn, are refilled from the second row, so each barrel here loses half its contents. And these barrels are filled from the top barrel, which thus becomes empty. This top barrel is then filled with the newest sherry for aging.[1]

The beauty of the solera system, is that if it is carefully managed, the bottom barrels will always contain an old mixture which can be sold. And this is constantly renewed. This system thus provides a continually renewed aged mixture.

Of course there’s much more to a solera system than this. A great deal spends on the skill of the cellar master, first to ensure that all barrels in the criaderas are maintained at optimum temperature and humidity, and then to ensure that the transfer of sherry between barrels is done in such a way so as not to disturb the maturing sherry.

[1] A very good explanation, with solera animations beginning at 7:37, can be found at https://bit.ly/3r7Nh4Z.

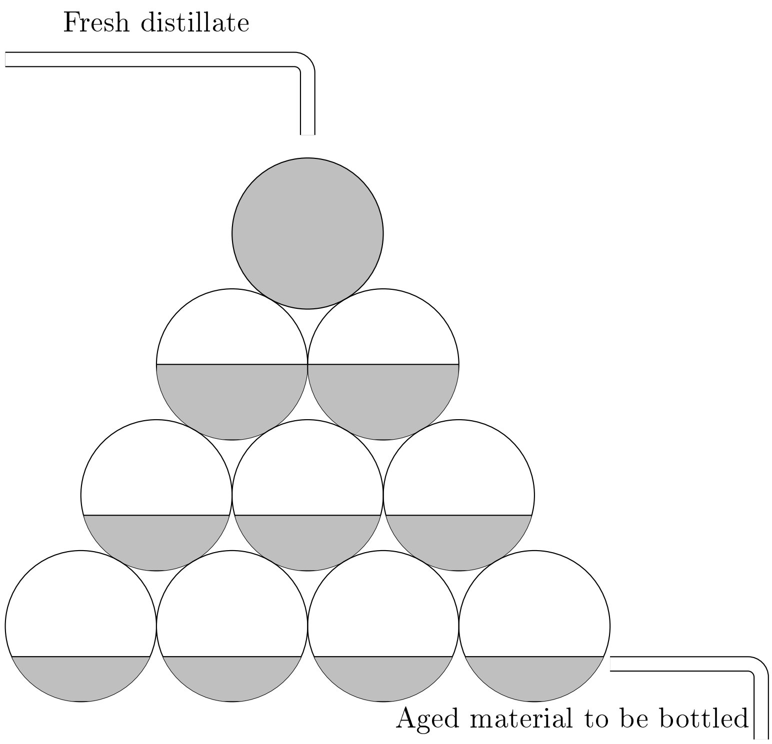

Figure 1: The solera system

This system is now in use world-wide to manufacture fortified wine and spirits of all sorts, as well as condiments such as balsamic vinegar.

2 Basic modelling

Before we start, note that there are several articles already about fractional blending; for example [1, 3, 6]. None of these adopt the approach we've taken, nor do they seem to use the same mathematical model to describe the blending. Moreover, none of them use a CAS (the first two were written long before and CAS was available), and as we shall see, the use of a CAS allows us to work with a simple model, and to manage with ease the expressions that arrive in the course of our use.

To develop a model, we shall consider four barrels of different volumes, which at aging time n will have ages a_n, b_n, c_n, d_n. In the four layer system (as in figure 1), one quarter of d_n is taken off and replaced from c_n. Thus immediately after this transfer, the age of the mixture in d is

$$\frac{3}{4}d_n + \frac{1}{4}c_n$$

At the end of the next aging period, before the next transfer, this mixture will have aged by one period which produces

$$d_{n+1} = \frac{3}{4}d_n + \frac{1}{4}c_n + 1.$$

Likewise we have

$$c_{n+1} = \frac{2}{3}c_n + \frac{1}{3}b_n + 1$$
$$b_{n+1} = \frac{1}{2}b_n + \frac{1}{2}a_n + 1$$
$$a_{n+1} = 1.$$

The last equation is because the barrel on the top row is emptied and filled afresh. What we have now is a system of *difference equations* [5, 2] relating the ages at stage n to the ages at the previous stage.

Suppose at the end of the first periods, before any transfer is done, we have

$$a(1) = 1, \quad b(1) = 2, \quad c(1) = 3, \quad d(1) = 4.$$

To avoid too much tangled algebra, our tool of choice will be Python's SymPy [4] module, which provides an `rsolve` command for solving difference equations. To do this, first rewrite using integers only:

$$2b_{n+1} = b_n + a_n + 2$$

and use $a_n = 1$ as shown in Listing **??**.

```
In[]: import sympy as sy
In[]: b = sy.Function('b')
In[]: c = sy.Function('b')
In[]: d = sy.Function('b')

In[]: bn = sy.rsolve(2*b(n+1)-b(n)-3,b(n),{b(1):2}); bn
```

$$3 - \frac{2}{2^n}$$

```
In[]: cn = sy.rsolve(3*c(n+1)-2*c(n)-bn-3,c(n),{c(1):3}); cn
```

$$-\frac{15}{2}\left(\frac{2}{3}\right)^n + \frac{4}{2^n} + 6$$

```
In[]: dn = sy.rsolve(4*d(n+1)-3*d(n)-cn-4,d(n),{d(1):4}); dn
```

$$\frac{45}{2}\left(\frac{2}{3}\right)^n - \frac{76}{3}\left(\frac{3}{4}\right)^n - \frac{4}{2^n} + 10$$

This means that the asymptotic age of the solera barrels will be 10 aging periods. If the transfer of material between the barrels takes place every six months, the age in the bottom barrels will approach 5 years.

We can work with the recurrence relations to obtain individual equations for each one. We start with

$$2b_{n+1} = b_n + 3 \quad \Rightarrow \quad 2b_{n+1} - b_n = 3 \tag{1}$$

and we write out the recurrence relation for c_n twice:

$$3c_{n+1} = 2c_n + b_n + 3 \tag{2}$$
$$3c_{n+2} = 2c_{n+1} + b_{n+1} + 3 \tag{3}$$

Now suppose we multiply the equation (3) by 2 and subtract equation (2) from it:

$$6c_{n+2} - 3c_{n+1} = 4c_{n+1} - 2c_n + 2b_{n+1} - b_n + 2(3) - 3.$$

Bu equation 1 we can replace $2b_{n+1} - b_n$ with 3, thus producing

$$6c_{n+2} - 3c_{n+1} = 4c_{n+1} - 2c_n + 6$$

This can be cleaned up to produce

$$6c_{n+2} - 7c_{n+1} + 2c_n = 6. \tag{4}$$

The same thing can be done to produce a recurrence relation for d_n, writing it out three times and eliminating the c_n terms by equation (4):

$$4d_{n+1} = 3d_n + c_n - 4 \tag{5}$$
$$4d_{n+2} = 3d_{n+1} + c_{n+1} - 4 \tag{6}$$
$$4d_{n+3} = 3d_{n+2} + c_{n+2} - 4 \tag{7}$$

In this case, to use equation (4) we compute

$$6 \times (7) - 7 \times (6) + 2 \times (5)$$

This will eliminate all the c terms, replacing them with 6, and will produce:

$$24d_{n+3} - 28d_{n+2} + 8d_{n+1} = 18d_{n+2} - 21d_{n+1} + 6d_n + 6 + 4$$

This can be rewritten as

$$24d_{n+3} - 46d_{n+2} + 29d_{n+1} - 6d_n = 10 \tag{8}$$

The coefficients in equations (4) and (8) may look at first to be quite random, but this is not the case. We first notice that the characteristic equations for each recurrence relation are easily factorized into linear factors:

$$6\lambda^2 - 7\lambda + 2 = (2\lambda - 1)(3\lambda - 2)$$
$$24\lambda^3 - 28\lambda^2 + 29\lambda - 6 = (2\lambda - 1)(3\lambda - 2)(4\lambda - 3)$$

and this pattern can be continued.

Also, if we create an array A of the coefficients of each relation:

	1	2	3	4	5
1	1				
2	2	-1			
3	6	-7	2		
4	24	-46	29	-6	
5	120	-326	329	-146	24

it is easy to see that (assuming all empty cells to have zero values):

$$A_{n,1} = n!, \quad A_{n,k} = nA_{n-1,k} - (n-1)A_{n-1,k-1} \quad \text{for} \quad k \geq 1.$$

3 Matrix formulation

The difference equations can be written in matrix form as

$$\begin{bmatrix} b_{n+1} \\ c_{n+1} \\ d_{n+1} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/3 & 2/3 & 0 \\ 0 & 1/4 & 3/4 \end{bmatrix} \begin{bmatrix} b_n \\ c_n \\ d_n \end{bmatrix} + \begin{bmatrix} 3/2 \\ 1 \\ 1 \end{bmatrix}$$

We don't include an equation for a_n as that is constant. Writing

$$\mathbf{b}_{n+1} = A\mathbf{b}_n + X$$

and with a starting vector $\mathbf{b}_1$, we have

$$\mathbf{b}_{n+1} = A^n\mathbf{b_1} + (A^{n-1} + A^{n-2} + \cdots + A^2 + A + I)X$$

and the sum of powers of A can be written as

$$(A^n - I)(A - I)^{-1}.$$

and so the expression for $\mathbf{b}_{n+1}$ is:

$$\mathbf{b}_{n+1} = A^n\mathbf{b}_1 + (A^n - I)(A - I)^{-1}X. \tag{9}$$

The matrix A is diagonalizable:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 2/3 & 0 \\ 0 & 0 & 3/4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 2 & 3 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 2/3 & 0 \\ 0 & 0 & 3/4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

Thus

$$A^n = \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} (1/2)^n & 0 & 0 \\ 0 & (2/3)^n & 0 \\ 0 & 0 & (3/4)^n \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (1/2)^n & 0 & 0 \\ -2(1/2)^n + 2(2/3)^n & (2/3)^n & 0 \\ 2(1/2)^n - 6(2/3)^n + 4(3/4)^n & -3(2/3)^n + 3(3/4)^n & (3/4)^n \end{bmatrix}$$

Given the matrix A^n we can now develop the rest of the result, noting that

$$(A - I)^{-1} = \begin{bmatrix} -2 & 0 & 0 \\ -2 & -3 & 0 \\ -2 & -3 & -4 \end{bmatrix}$$

and using equation (9). This produces

$$\begin{bmatrix} b_{n+1} \\ c_{n+1} \\ d_{n+1} \end{bmatrix} = \begin{bmatrix} 3 - \left(\frac{1}{2}\right)^n \\ 6 + 2\left(\frac{1}{2}\right)^n - 5\left(\frac{2}{3}\right)^n \\ 10 - 2\left(\frac{1}{2}\right)^n + 15\left(\frac{2}{3}\right)^n - 19\left(\frac{3}{4}\right)^n \end{bmatrix}$$

To get the equations for b_n, c_n and d_n note that, for example, $(2/3)^{n-1} = (3/2)(2/3)^n$. Thus all powers can be scaled to produce powers one less. The result is:

$$\begin{aligned} b_n &= 3 - 2\left(\frac{1}{2}\right)^n \\ c_n &= 6 + 4\left(\frac{1}{2}\right)^n - \frac{15}{2}\left(\frac{2}{3}\right)^n \\ d_n &= 10 - 4\left(\frac{1}{2}\right)^n + \frac{45}{2}\left(\frac{2}{3}\right)^n - \frac{76}{3}\left(\frac{3}{4}\right)^n \end{aligned}$$

and it will be seen that these are the same equations as obtained earlier on using the `rsolve` command in Python's SymPy module.

All of the above can be simplified by writing the recurrence as:

$$\begin{bmatrix} b_{n+1} \\ c_{n+1} \\ d_{n+1} \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 & 3/2 \\ 1/3 & 2/3 & 0 & 1 \\ 0 & 1/4 & 3/4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_n \\ c_n \\ d_n \\ 1 \end{bmatrix}$$

The matrix M can also be diagonalized as $M = ADA^{-1}$:

$$\begin{bmatrix} 1/2 & 0 & 0 & 3/2 \\ 1/3 & 2/3 & 0 & 1 \\ 0 & 1/4 & 3/4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ -2 & -1 & 0 & 6 \\ 2 & 3 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 \\ 0 & 0 & 3/4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 3 \\ -2 & -1 & 0 & 6 \\ 2 & 3 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

Since the top three diagonal elements of D are all less than one, the limiting value of D^n will be a 4×4 matrix all zero except for the bottom right element, which is 1. This means that the limiting values of the ages are:

$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ -2 & -1 & 0 & 6 \\ 2 & 3 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 3 \\ -2 & -1 & 0 & 6 \\ 2 & 3 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \\ 1 \end{bmatrix}$$

This approach also shows that the limiting value is independent of the starting ages, since

$$\begin{bmatrix} 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ c_0 \\ d_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \\ 1 \end{bmatrix}.$$

4 Limiting values

As n increases, the age in the barrels at each layer approaches a constant value. We diagonalize A as PDP^{-1} and since every diagonal element of D is less than one, it follows that A^n approaches zero. From equation (9) if A^n is set to zero, we have, as a limiting vector:

$$-(A - I)^{-1}X = -\begin{bmatrix} -2 & 0 & 0 \\ -2 & -3 & 0 \\ -2 & -3 & -4 \end{bmatrix} \begin{bmatrix} 3/2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix}$$

To generalize this, suppose we have k rows, with A being the matrix whose diagonal elements will be 1/2, 2/3, 3/4, 4/5, ... and whose sub-diagonal elements will be 1/3, 1/4, 1/5, ..., like this:

$$A = \begin{bmatrix} 1/2 & & & & & \\ 1/3 & 2/3 & & & & \\ & 1/4 & 3/4 & & & \\ & & 1/5 & 4/5 & & \\ & & & \ddots & \ddots & \end{bmatrix}$$

Then:

$$A - I = \begin{bmatrix} 1/2 & & & & & \\ 1/3 & -1/3 & & & & \\ & 1/4 & -1/4 & & & \\ & & 1/5 & -1/5 & & \\ & & & \ddots & \ddots & \end{bmatrix}$$

Hand or computer-aided computation (for example, the Python command `(A-sy.eye(3)).inv()`) can be used to show that for increasing sizes, the matrices $(A-I)^{-1}$ are:

$$\begin{bmatrix} -2 & 0 & 0 \\ -2 & -3 & 0 \\ -2 & -3 & -4 \end{bmatrix}, \quad \begin{bmatrix} -2 & 0 & 0 & 0 \\ -2 & -3 & 0 & 0 \\ -2 & -3 & -4 & 0 \\ -2 & -3 & -4 & -5 \end{bmatrix}, \quad \begin{bmatrix} -2 & 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & 0 & 0 \\ -2 & -3 & -4 & 0 & 0 \\ -2 & -3 & -4 & -5 & 0 \\ -2 & -3 & -4 & -5 & -6 \end{bmatrix}$$

This pattern is easily proved by induction. Suppose that A_k is the $k \times k$ version of $A - I$, and B_k is the inverse. Then we can define A_{k+1} and B_{k+1} with block matrices:

$$A_{k+1} = \left[\begin{array}{c|c} A_k & 0_k \\ \hline 0 \quad 0\ldots0 \quad \frac{1}{k+1} & -\frac{1}{k+1} \end{array}\right], \qquad B_{k+1} = \left[\begin{array}{c|c} B_k & 0_k \\ \hline -2\ -3\ -4\ldots-k & -(k+1) \end{array}\right]$$

writing u_{k+1} for the bottom left block of A_{k+1}, and v_{k+1} for the bottom left block of B_{k+1}, their product is

$$A_{k+1}B_{k+1} = \left[\begin{array}{c|c} A_kB_k + 0_kv_{k+1} & A_k0_k + 0_k[-(k+1)] \\ \hline u_kB_k + (-\frac{1}{k+1})v_k & u_k0_k + (-\frac{1}{k+1})(-(k+1)) \end{array}\right]$$

All cells are easily computed, but the bottom left cell needs a little explanation. We note that u_k consists of only one non-zero value $1/(k+1)$, in the last place, so that u_kB_k essentially multiples the last row of B_k by that non-zero value. But the last row of B_k and v_k are the same vector, so the bottom left cell is

$$\frac{1}{k+1}v_k + \left(-\frac{1}{k+1}\right)v_k = 0.$$

Thus

$$A_{k+1}B_{k+1} = \left[\begin{array}{c|c} I & 0_k \\ \hline 0_k^T & 1 \end{array}\right]$$

which is the $(k+1) \times (k+1)$ identity matrix.

Since the limiting age of the barrels in the solera row has been shown to be

$$-(A-I)^{-1}X$$

where X is a column vector starting with 3/2 but all other values are 1's. This is then equal to

$$-\begin{bmatrix} -2 & & & & \\ -2 & -3 & & & \\ -2 & -3 & -4 & & \\ \vdots & & & & \\ -2 & -3 & -4 & \cdots & -k \end{bmatrix} \begin{bmatrix} 3/2 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3+3 \\ 3+3+4 \\ \vdots \\ 3+3+4+\ldots+k \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \\ \vdots \\ \frac{k^2+k}{2} \end{bmatrix}$$

The final value shows that the limiting age of the solera barrels in a system with k layers is $(k^2+k)/2$.

5 Other fractions

Up until now, we have been moving one compete barrel between rows, thus leaving the top barrel completely empty each time. But suppose we move some fraction p of a barrel instead, where $0 < p \leq 1$.

As before, we'll start with a four-tier system. Emptying a total of p barrels from the solera reduces each barrel by $p/4$. This amount needs to be replaced from the third row, thus reducing each of those barrels by $p/3$. Similarly, $p/2$ barrels will be taken from each of the barrels in the second row, and as described at the beginning of this section, p from the top barrel.

Since the top barrel is not necessarily completely emptied, it will need to be considered. The recurrence relation relating ages in each barrel are then:

$$\begin{aligned} a_{n+1} &= (1-p)a_n + 1 \\ b_{n+1} &= (1-\frac{p}{2})b_n + \frac{p}{2}a_n + 1 \\ c_{n+1} &= (1-\frac{p}{3})c_n + \frac{p}{3}b_n + 1 \\ d_{n+1} &= (1-\frac{p}{4})d_n + \frac{p}{4}c_n + 1 \end{aligned}$$

If $p = 1$ these equations are equal to our original equations.

It is in fact easy to solve these using matrix methods, with

$$\begin{bmatrix} a_{n+1} \\ b_{n+1} \\ c_{n+1} \\ d_{n+1} \end{bmatrix} = \begin{bmatrix} 1-p & 0 & 0 & 0 \\ \frac{p}{2} & 1-\frac{p}{2} & 0 & 0 \\ 0 & \frac{p}{3} & 1-\frac{p}{3} & 0 \\ 0 & 0 & \frac{p}{4} & 1-\frac{p}{4} \end{bmatrix} \begin{bmatrix} a_n \\ b_n \\ c_n \\ d_n \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

If we write this as

$$\mathbf{x}_{n+1} = A\mathbf{x}_n + \mathbf{b}$$

then, as before

$$\mathbf{x}_n = A^n\mathbf{x}_0 + (A^n - I)(A-I)^{-1}\mathbf{b} \tag{10}$$

and so the solution reduces to finding A^n. But again we have an easily diagonalizable matrix, with

$$\begin{bmatrix} 1-p & 0 & 0 & 0 \\ 1-\frac{p}{2} & \frac{p}{2} & 0 & 0 \\ 0 & 1-\frac{p}{3} & \frac{p}{3} & 0 \\ 0 & 0 & 1-\frac{p}{4} & \frac{p}{4} \end{bmatrix}$$
$$= \begin{bmatrix} -6 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ -3 & -2 & -1 & 0 \\ 1 & 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1-p & 0 & 0 & 0 \\ 0 & 1-\frac{p}{2} & 0 & 0 \\ 0 & 0 & 1-\frac{p}{3} & 0 \\ 0 & 0 & 0 & 1-\frac{p}{4} \end{bmatrix} \begin{bmatrix} -6 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ -3 & -2 & -1 & 0 \\ 1 & 2 & 3 & 1 \end{bmatrix}^{-1}$$

Also,

$$(A-I)^{-1} = \begin{bmatrix} -\frac{1}{p} & 0 & 0 & 0 \\ -\frac{1}{p} & -\frac{2}{p} & 0 & 0 \\ -\frac{1}{p} & -\frac{2}{p} & -\frac{3}{p} & 0 \\ -\frac{1}{p} & -\frac{2}{p} & -\frac{3}{p} & -\frac{4}{p} \end{bmatrix}$$

Substituting into equation (10), and with $a_0 = 1$, $b_0 = c_0 = d_0 = 0$ produces

$$\begin{aligned}
a_n &= \left(1 - \frac{1}{p}\right)(1-p)^n + \frac{1}{p} \\
b_n &= \left(-1 + \frac{1}{p}\right)(1-p)^n + \left(1 - \frac{4}{p}\right)\left(1 - \frac{p}{2}\right)^n + \frac{3}{p} \\
c_n &= \left(\frac{1}{2} - \frac{1}{2p}\right)(1-p)^n + \left(-2 + \frac{8}{p}\right)\left(1 - \frac{p}{2}\right)^n + \left(\frac{3}{2} - \frac{27}{2p}\right)\left(1 - \frac{p}{3}\right)^n + \frac{6}{p} \\
d_n &= \left(-\frac{1}{6} + \frac{1}{6p}\right)(1-p)^n + \left(2 - \frac{8}{p}\right)\left(1 - \frac{p}{2}\right)^n + \left(-\frac{9}{2} + \frac{81}{2p}\right)\left(1 - \frac{p}{3}\right)^n \\
&\quad + \left(\frac{8}{3} - \frac{128}{3p}\right)\left(1 - \frac{p}{4}\right)^n + \frac{10}{p}
\end{aligned}$$

We can see that the limiting values in the solera row, in a criederas with k layers, will be

$$\frac{k^2+k}{2p}.$$

And this can be easily shown by noting the general form of $(A-I)^{-1}$, which can be established by induction. Let A_k be the matrix $A - I$ of size $k \times k$. Then we can express A_{k+1} as a block matrix:

$$A_{k+1} = \left[\begin{array}{cccc|c} & & A_k & & 0_k \\ \hline 0 & 0 & 0\dots0 & \frac{p}{k+1} & -\frac{p}{k+1} \end{array}\right]$$

If B_k is the inverse of A_k, then

$$B_{k+1} = \left[\begin{array}{cccc|c} & & B_k & & 0_k \\ \hline -\frac{1}{p} & -\frac{2}{p} & -\frac{3}{p} \cdots & -\frac{k}{p} & -\frac{k+1}{p} \end{array}\right]$$

Writing the lower left block of A_{k+1} as u_k and of B_{k+1} as v_k, we have:

$$A_{k+1}B_{k+1} \left[\begin{array}{c|c} A_kB_k + 0_kv_k & A_k0_k + 0_k(-\frac{k+1}{p}) \\ \hline u_kB_k + (-\frac{p}{k+1})v_k & u_k0_k + (-\frac{p}{k+1})(-\frac{k+1}{p}) \end{array}\right]$$

By the induction hypothesis, $A_kB_k = I$. For the bottom left, we note that by construction, v_k equals the bottom row of B_k. And since u_k consists of zeros except for a final value of $p/(k+1)$, the results of the first product is simply this value multiplied into every element of the last row of B_k. Thus:

$$\begin{aligned} u_kB_k + \left(-\frac{p}{k+1}\right)v_k &= \left(\frac{p}{k+1}\right)\left[-\frac{1}{p} \quad -\frac{2}{p} \quad -\frac{3}{p} \cdots -\frac{k}{p}\right] \\ &\quad + \left(\frac{p}{k+1}\right)\left[-\frac{1}{p} \quad -\frac{2}{p} \quad -\frac{3}{p} \cdots -\frac{k}{p}\right] \\ &= 0 \end{aligned}$$

All other products are more straightforward; in the end we have

$$A_{k+1}B_{k+1} = \left[\begin{array}{ccc|c} & I & & 0_k \\ \hline 0 & 0 & 0 \ldots 0 & 1 \end{array}\right]$$

which is the identity, as required.

The limiting value of the age in the solera row is then

$$-(A - I)^{-1}\mathbf{b}$$

and since $\mathbf{b}$ consists entirely of 1's, this product is

$$\begin{bmatrix} \frac{1}{p} \\ \frac{1}{p} + \frac{2}{p} \\ \frac{1}{p} + \frac{2}{p} + \frac{3}{p} \\ \vdots \\ \frac{1}{p} + \frac{2}{p} + \frac{3}{p} + \cdots + \frac{k}{p} \end{bmatrix} = \begin{bmatrix} \frac{1}{p} \\ \frac{3}{p} \\ \frac{6}{p} \\ \vdots \\ \frac{k^2+k}{2p} \end{bmatrix}$$

6 General arithmetic sequences

In the previous section, the number of barrels in row k (starting with the top row numbered 1) is k. We now consider a more general system where the k-row contains $g + (k-1)h$ barrels, where $g,h \geq 1$ (and are integers). And at each stage m barrels are moved into the top row and between other rows, with $m \leq g$.

In the first row, assuming all barrels to be full, m barrels are moved to row 2, and $g-m$ barrels are left. We are thus moving a fraction of m/g material between rows. But in this more general situation, we have $1-m/g$ material left in each of the top barrels. This will then be aged between periods, so that:

$$a_{n+1} = \left(1-\frac{m}{g}\right)a_n + 1. \tag{11}$$

Here the fraction m/g corresponds to the fraction p in the previous section. This means that we can re-purpose the equations given at the beginning of section 5 to obtain the other equations:

$$\begin{aligned} b_{n+1} &= \left(1-\frac{m}{g+h}\right)b_n + \frac{m}{g+h}a_n + 1 \\ c_{n+1} &= \left(1-\frac{m}{g+2h}\right)c_n + \frac{m}{g+2h}b_n + 1 \\ d_{n+1} &= \left(1-\frac{m}{g+3h}\right)d_n + \frac{m}{g+3h}c_n + 1 \end{aligned} \tag{12}$$

Multiplying out to clear the fractions produces:

$$\begin{aligned} ga_{n+1} &= (g-m)a_n + ma_{n-1} + g \\ (g+h)b_{n+1} &= (g+h-m)b_n + ma_n + g + h \\ (g+2h)c_{n+1} &= (g+2h-m)c_n + mb_n + g + 2h \\ (g+3h)d_{n+1} &= (g+3h-m)d_n + mc_n + g - 3h \end{aligned}$$

The first two can be entered into SymPy as:

```
In[]: n,m,g,h = sy.var('n,m,g,h')
In[]: an = sy.rsolve(g*a(n+1)-(g-m)*a(n)-g,a(n),{a(1):1})
In[]: bn = sy.rsolve((g+h)*b(n+1)-(g+h-m)*b(n)-sy.simplify(m*an)-g-h,\
      b(n),{b(1):0})
```

These turn out to have the splendid solutions:

$$a_n = -\frac{g\left(\frac{g-m}{g}\right)^n}{m} + \frac{g}{m}$$

$$b_n = \frac{\left(\frac{g+h-m}{g+h}\right)^n\left(-g^3-3g^2h+g^2m-3gh^2+ghm-h^3\right)}{ghm+h^2m-hm^2} + \frac{g^2\left(\frac{g-m}{g}\right)^n + h\left(2g+h\right)}{hm}$$

With a little bit of algebra (helped by SymPy) this last can be written as:

$$\begin{aligned} b_n &= -\frac{(g+h)(g^2+2gh-gm+h^2)}{hm(g+h-m)}\left(\frac{g+h-m}{g+h}\right)^n + \frac{g^2}{hm}\left(\frac{g-m}{g}\right)^n + \frac{2g+h}{m} \\ &= \frac{g^2+2gh-gm+h^2}{hm}\left(\frac{g+h-m}{g+h}\right)^{n-1} + \frac{g^2}{hm}\left(\frac{g-m}{g}\right)^n + \frac{2g+h}{m} \end{aligned}$$

Already we see that we're obtaining expressions of considerable complexity. It turns out that the mechanisms of SymPy are unable to solve the comparable equation of c_n directly, but we can give it some help.

First note that the form of the difference equation for c_n is

$$c_{n+1} = \left(1 - \frac{m}{g+2h}\right) c_n - \frac{m}{g+2h}\left(A\left(1-\frac{m}{g}\right)^n + B\left(1-\frac{m}{g+h}\right)^n + C\right) - 1$$

where A, B, C are the coefficients from b_n. We can write this more simply as:

$$c_{n+1} = tc_n - Ax^n - By^n - C - 1$$

where A, B, C now include the multiplier $m/(g+h)$. This can be solved:

```
In[]: from sympy.abc import t,x,y,A,B,C
In[]: cn = sy.rsolve(c(n+1)-t*c(n)-A*x**n-B*y**n-1,c(n),{c(1):0})
```

The output is too long to be shown, but all we need to is to extract from the expression for b_n the papers corresponding to A, B, C, X, y and substitute them into the expression for c_n just obtained, along with the definition for t.

We have seen that the expression for b_n is reasonably hideous, but the coefficients can be extracted by setting various of the powers to zero with a little function:

```
In[]: def power_sub(i,j):
      ....temp = {((g-m)/g)**n:i,((g+h-m)/(g+h))**n:j})
      ....return(temp)

In[]: C1 = bn.subs(power_sub(0,0))
In[]: A1 = (bn - C1).subs(power_sub(1,0))
In[]: B1 = (bn - C1).subs(power_sub(0,1))
```

Here `C1` is the constant term of b_n, and `A1`, `B1` are the coefficients of

$$\left(\frac{g-m}{g}\right)^n, \quad \left(\frac{g+h-m}{g+h}\right)^n$$

respectively. Finally all of this can be put into the expression for c_n:

```
In[]: x1 = (g-m)/g
In[]: y1 = (g+h-m)/(g+h)
In[]: t1 = (g + 2*h - m)/(g + 2*h)
In[]: A1 *= m/(g+2*h)
In[]: B1 *= m/(g+2*h)
In[]: C1 *= m/(g+2*h)
In[]: cn1 = cn.subs({A:A1,B:B1,C:C1,x:x1,y:y1,t:t1})
```

This is still a very complicated expression, but we can get a sense of it by noting that it will have the form:

$$c_n = C_3\left(\frac{g+2h-m}{g+2h}\right)^n + C_2\left(\frac{g+h-m}{g+h}\right)^n + C_1\left(\frac{g-m}{g}\right)^n + C_0 \tag{13}$$

As for b_n above, we can find the values of the coefficients C_k by setting various of the powers to zero, again with a small function:

```
In[]: def power_sub3(i,j,k):
      ....temp = {((g-m)/g)**n:i,((g+h-m)/(g+h))**n:j,
                  ((g+2*h-m)/(g+2*h))**n:k}
      ....return(temp)

In[]: C0 = sy.simplify(cn2.subs(power_sub3(0,0,0)))
In[]: C1 = sy.simplify((cn2-C0).subs(power_sub3(1,0,0)))
In[]: C2 = sy.simplify((cn2-C0).subs(power_sub3(0,1,0)))
In[]: C3 = sy.simplify((cn2-C0).subs(power_sub3(0,0,1)))
```

These are:

$$C_0 = \frac{3(g+h)}{m}$$
$$C_1 = \frac{2g+h}{g+2h}$$
$$C_2 = \frac{g^4+4g^3h-g^3m+6g^2h^2-2g^2hm+4gh^3-gh^2m+h^4}{h^2m\,(g+h-m)}$$
$$C_3 = \frac{-g^4-8g^3h+g^3m-24g^2h^2+4g^2hm-32gh^3+4gh^2m-16h^4}{2h^2m\,(g+2h-m)}$$

Substituting these in equation 13 will provide the full solution for c_n. Note that since all the powers are of values less than 1, the limiting value as n increases is the constant term.

We can now turn our attention to the fourth row, given as the solution to the difference equation for d_n. However, given the complexities in trying to solve the previous equation for c_n, we will not try (although we could, if we felt like giving ourselves a hard time), but go straight to determining the limit.

The equation is

$$d_n = \left(1-\frac{m}{g+3h}\right)d_n + \frac{m}{g+3h}c_n + 1$$

and its solution will have the form

$$d_n = A_1\left(\frac{g+3h-m}{g+3h}\right)^n + A_2\left(\frac{g+2h-m}{g+2h}\right)^n + A_3\left(\frac{g+h-m}{g+h}\right)^n + A_4\left(\frac{g-m}{g}\right)^n + A_5$$

Our only concern is to find A_5. We shall begin as we did for c_n above, but without aiming to determine any of the coefficients, simply set all the powers to zero.

```
In[]: A,B,C,D,x,y,z = sy.var('A,B,C,D,x,y,z')
In[]: dn = sy.rsolve((g+3*h)*d(n+1)-(g+3*h-m)*d(n)-\
      m*(A*x**n+B*y**n+C*z**n+D)-g-3*h,d(n),{d(0):0})
In[]: dn1 = sy.simplify(dn)
In[]: const_d = dn1.subs({((g+3*h-m)/(g+3*h))**n:0,x**n:0,y**n:0,z**n:0})
```

Here the expression $Ax^n + by^n + Cz^n + D$ stands for c_n; the values of A, b and C are irrelevant because each of x, y, z ae less than 1. This produces:

$$\frac{Dm+g+3h}{m}$$

But the D is in fact the constant term from c_n, and substituting this for D produces the limiting age for the material in the fourth row:

$$\frac{4g+6h}{m}.$$

Note that if we set $m = g = h = 1$ corresponding to the original setup, then this value is equal to 10 (as it should).

The limiting values in the first four rows are:

$$\begin{aligned} a_\infty &= \frac{g}{m} \\ b_\infty &= \frac{2g+h}{m} \\ c_\infty &= \frac{3g+3h}{m} \\ d_\infty &= \frac{4g+6h}{m}. \end{aligned}$$

A pattern is now apparent, and can be continued for increasing n. Note that if all variables are equal to 1, these limiting values are 1, 3, 6, 10, as previously.

We can demonstrate this by considering the general solution of a difference equation, for example

$$x_{n+1} = \lambda x_n + pz^n + qw^n + r.$$

Assuming that all of λ, z, w are distinct, and using the method of undetermined coefficients, then the homogeneous solution will be of the form $x_n = A\lambda^n$ and the particular solution of the form $y_n = Pz^n + Qw^n + R$.

Substituting y_n into the difference equation produces

$$Pz^{n+1} + Qw^{n+1} + R = \lambda(Pz^n + Qw^n + R) + pz^n + qw^n + r.$$

We can now equate the coefficients of z^n, w^n, and the constant term on both sides; these equations are, respectively:

$$\begin{aligned} Pz &= \lambda P + p \\ Qw &= \lambda Q + q \\ R &= \lambda R + r \end{aligned}$$

It is the last value which concerns us, and so

$$R = \frac{r}{1-\lambda}.$$

This is clearly generalizable to particular solutions of any length. In our context, all values λ, z, w are less then 1, so that in the limit only the constant term matters.

Going back to the original set of difference equations 11 and 12 we have that the constant term for a_n is g/m.

This means that the constant term in the equation for b_n is

$$r = \frac{m}{g+h}\left(\frac{g}{m}\right) + 1 = \frac{g}{g+h} + 1.$$

For this equation,

$$\lambda = 1 - \frac{m}{g+h}$$

and so

$$\frac{r}{1-\lambda} = \frac{2g+h}{m}$$

as we found previously. For the next equation (for c_n), we have

$$r = \frac{m}{g+2h}\left(\frac{2g+h}{m}\right) + 1 = \frac{2g+h}{g+2h} + 1$$

Since here, $\lambda = 1 - m/(g+2h)$, we have

$$\frac{r}{1-\lambda} = \frac{3g+3h}{m}.$$

In general, suppose that k_i is the constant term corresponding to the i-th row. We then have

$$\begin{aligned} k_i &= \left(\frac{m}{g+(i-i)h}k_{i-1} + 1\right) \Big/ \left(\frac{m}{g+(i-1)h}\right) \\ &= k_{i-1} + \frac{g+(i-1)h}{m}. \end{aligned}$$

Since $k_1 = g/m$, we have, in general,

$$k_i = \frac{ig + i(i-1)h/2}{m} = \frac{2ig + i(i-1)h}{2m} = \frac{i}{2m}(2g + (i-1)h).$$

7 A final generalization

Suppose now that the numbers of barrels in row i is g_i, where these values form a non-decreasing sequence.

$$1 \leq g_1 \leq g_2 \leq g_3 \leq \ldots$$

Again we use m for the number of barrels-worth of liquid moved each time. We won't attempt to try and find solutions to the difference equations, but simply determine the constant values for each row. As before, the first constant value will be

$$k_1 = \frac{g_1}{m}.$$

A difference equation, say for row 4, will look like this:

$$d_{n+1} = \left(1 - \frac{m}{g_4}\right) d_n + \frac{m}{g^4} c_n + 1$$

so that in all equations, $g + (i-1)h$ will be replaced with g_i. Then for row $i > 1$, we will have

$$\begin{aligned} k_i &= \left(\frac{m}{g_i}k_{i-1} + 1\right) \Big/ \left(\frac{m}{g_i}\right) \\ &= k_{i-1} + \frac{g_i}{m}. \end{aligned}$$

This means, for example, that

$$k_4 = \frac{g_1 + g_2 + g_3 + g_4}{m}$$

and so in general,

$$k_n = \frac{1}{m} \sum_{i=1}^{n} g_i.$$

This generalizes the results of previous sections.

8 Final remarks

Although this work was inspired by a "real-life" situation (the visit of the author to a friend's winery and distillery), it is in fact a nice example of the use of a computer algebra system to solve systems of difference equations, both by standard techniques and also by the use of matrix algebra. Even when the complexities were such that individual solutions could not be (easily) obtained, we showed how to find the limiting values, which is in fact what counts in the world of fine wines and spirits. As a multi-billion dollar industry covering much of the world, this is clearly an industry worth considering, at least mathematically. We note that fractional blending is most often used for alcohol, and so might seem of no interest to people who don't or can't drink alcohol. However, it is also used for non-alcoholic purposes such as for fine balsamic vinegars. We thus note that fractional blending systems have a very wide usage, and the science of them may thus have a wide appeal.

Acknowledgements

The author gratefully acknowledges the anonymous reviewers who provided detailed information about various errors, inconsistencies, and typos in the first draft of this article. These have been amended, and any errors that remain are the author's responsibility.

References

[1] G Baker, M Amerine, E Roessler, et al. "Theory and application of fractional-blending systems". In: *Hilgardia* 21.14 (1952), pp. 383–409.

[2] Paul Cull, Mary Flahive, and Robby Robson. *Difference Equations: From Rabbits To Chaos*. Springer, 2005.

[3] F Javier Girón and M Lina Martínez. "Asymptotic Behavior of Fractional Blending Systems". In: *The Mathematics of the Uncertain: A Tribute to Pedro Gil* (2018), pp. 185–194.

[4] Aaron Meurer et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (Jan. 2017), e103. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.103. URL: https://doi.org/10.7717/peerj-cs.103.

[5] Ronald E Mickens. *Difference Equations*. 3rd edition. CRC Press, 2015.

[6] RE Purser. "Fractional-Blending System: Age and Blend of Wine Produced". In: *American Journal of Enology and Viticulture* 18.4 (1967), pp. 175–181.

[7] Jancis Robinson. *The Oxford Companion to Wine*. 3rd edition. Oxford University Press, 2014.