On Morley’s theorem, determinantal identities and the $QR$ algorithm

A J Macfarlane
Centre for Mathematical Sciences, D.A.M.T.P., Wilberforce Road, Cambridge CB3 0WA, UK
E-mail: a.j.macfarlane@damtp.cam.ac.uk

Abstract. Three results are shown in this contribution. First, the diagram used in the proof of Morley’s theorem is shown to provide a template for the creation of a closed solid figure. Second, it is shown how a proposition governing a class of determinants can be used to produce a rich supply of identities involving, e.g., Fibonacci and related families of numbers. Finally, it is shown in the case of an arbitrary $2 \times 2$ symmetric matrix $A$, how to obtain explicit expressions for all the elements of all the matrices involved in implementing the determination of the eigenvalues of $A$ by means of the $QR$ algorithm.

Received: 18 October 2009

1. Introduction
This article is based on material presented at the Symposium Quantum Groups, Foundations, Information held in honour of Tony Sudbery on the occasion of his retirement. Tony, one of my first PhD students in Cambridge, became and is a valued lifelong friend. It is therefore my pleasure to offer this contribution to his Festschrift.

The article consists of three sections. The first shows how the diagram used in the proof of Morley’s theorem provides a template for the creation of a closed solid figure. The second shows how a proposition governing a class of determinants can be used to produce a rich supply of identities involving, e.g., Fibonacci and related families of numbers. The third section shows, in the case of an arbitrary $2 \times 2$ symmetric matrix $A$, how to obtain explicit expressions for all the elements of all the matrices involved in implementing the determination of the eigenvalues of $A$ by means of the $QR$ algorithm.

2. The Morley tricorn
Morley’s theorem states that the points of intersection of adjacent trisectors of the angles of any triangle are the vertices of an equilateral triangle, known as Morley’s triangle. The content of the theorem is illustrated in Fig. 1. In Fig. 1 additional points $P, Q, R$ and dotted lines $DP, EQ, FR$ have been included for use in the construction of the Morley tricorn, to be
described. The points \( P, Q, R \) points are uniquely determined so that

\[
AQ = AR = s - a, \quad BR = BP = s - b, \quad CP = CQ = s - c,
\]

where \( 2s = a + b + c \) in standard notation.

Concise proofs of Morley’s theorem are given in [1, 2], and a good picture of other studies on the subject is to be found at P1199 of [3].

Fig. 1 was considered at the symposium as the high point of a sequence of demonstrations of how the diagrams used for proofs of theorems in the geometry of plane triangles can be folded (origami-wise: valley folds on dotted lines and mountain folds on solid ones) so as to realise three-dimensional entities. It was shown there how Fig. 1 can be regarded as a template for the creation of a unique rigid exactly closed solid, called [4] the Morley tricorn, with ten undistorted plane faces.

To understand the process, suppose cuts along the dotted lines \( FR, EQ, DP \) have been made. Then fold the triangles \( \Delta AQE \) and \( \Delta ARF \) up over \( AE \) and \( AF \) respectively to bring \( AQ \) and \( AR \) into coincidence. Let \( W_A \) denote the point of coincidence of \( Q \) and \( R \). This step yields a pyramid, or horn, with \( \Delta EFW_A \) providing the outline of its base, and vertex \( A \). Similarly pyramids, on \( \Delta FDW_B \) and \( \Delta DEW_C \) as base and vertices \( B \) and \( C \) respectively, can be formed. The points \( W_B \) and \( W_C \) are the points at which the respective pairs of points \( R \) and \( P \), and \( P \) and \( Q \), are thereby brought into coincidence. It is then easy to see that the
three pyramids can be folded up over $EF, FD, DE$ to bring $W_A, W_B, W_C$ into coincidence at a point $W$, with exact closure that produces a ‘three horned’ three-dimensional figure, or tricorn. Full geometric justification of this statement can be given.

In fact, it is not necessary to cut along any lines at all. The foldings of the three pyramids just described, can be performed simultaneously, and in fact easily, in a process in which inward (valley) folds along the three dotted lines are made. This produces the Morley tricorn, a figure with central pyramid $DEF W$, which has the Morley triangle as base and vertex $W$, and pyramidal horns with vertices $A, B, C$, standing on the outlined slant faces of the central pyramid.

3. Difference equations and determinantal identities

Let $S$ denote a sequence $\{y_0, y_1, y_2, \ldots\}$ of real quantities $y_n$ governed by a difference equation

$$y_n = ay_{n-1} + by_{n-2},$$

(2)

where $a, b, y_0, y_1$ are given real numbers.

Let $T$ denote a sequence of $n \times n$ determinants

$$T_n(x, y_0, y_1, a, b) =
\begin{vmatrix}
  y_1 & y_2 & y_3 & \cdots & y_{n-2} & y_{n-1} & y_n \\
  -x & y_1 & y_2 & \cdots & y_{n-3} & y_{n-2} & y_{n-1} \\
  0 & -x & y_1 & \cdots & y_{n-4} & y_{n-3} & y_{n-2} \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
  0 & 0 & 0 & \cdots & y_1 & y_2 & \vdots \\
  0 & 0 & 0 & \cdots & -x & y_1 & y_2 \\
  0 & 0 & 0 & \cdots & 0 & -x & y_1 \\
\end{vmatrix},$$

(3)

where $x$ is an indeterminate. We now quote a result stated and proved in [5].

Proposition 1: If the $y_n \in S$ are subject to (2), with $x, a, b, y_0, y_1$ all fixed real numbers, then the determinants $T_n = T_n(x, y_0, y_1, a, b) \in T$ are related by the difference equation

$$T_n = (y_1 + ax)T_{n-1} + bx(y_0 + x)T_{n-2}.$$  

(4)

At the symposium, the proposition was used to present a miscellany of determinantal identities involving the Fibonacci (and related families of) numbers in either the role of $S$ or $T$ or both. A paper covering such material [5] has now appeared. Here, one illustrative example is given, and the origin of the ideas in the theory of one-dimensional cellular automata [6] is outlined.

3.1. An example involving Fibonacci numbers

The Fibonacci numbers $F_n$ are such that

$$F_{n+2} = F_{n+1} + F_n, \quad F_0 = F_1 = 1,$$

(5)

and hence

$$F_n = \frac{\alpha^n - \beta^n}{\alpha - \beta}, \quad \alpha, \beta = \frac{1 \pm \sqrt{5}}{2}.$$  

(6)
For the example, take $S = \{ y_0 = -1, y_1 = 1, -1, 1, \ldots \}$ so that $y_{n+2} = y_n, a = 0, b = 1$. Then, by Proposition 1, $T_n = T_n(\beta, -1, 1, 0, 1) \in T$ satisfies

$$T_{n+2} = T_{n+1} + T_n,$$

in the case in which $x = \beta$. It follows easily that

$$T_n = F_n - \beta F_{n-1} = \alpha^{n-1},$$

as can be checked. The choice $x = \alpha$ leads similarly to $T_n = \beta^{n-1}$.

### 3.2. The origin of Proposition 1

A one dimensional cellular automaton involves a linear array of cells, usually displayed as squares, with a cell variable $x_n(t)$ attached to the $n$-th cell at time $t$. Each cell variable takes on the value 1 or 0, usually displayed as dark or light coloured square. Consider the situation in which the cellular automaton CA150, governed by Wolfram rule 150 [6], evolves from the initial state of 'single seed', for which $x_0(t = 0) = 1$ and $x_n(t = 0) = 0$ for $n \neq 0$. Time advances in discrete steps through values $t = 0, 1, 2, \ldots$, and the state at time $(t+1)$ of CA150 is determined by the state at time $t$ by rule 150, i.e.

$$x_n(t + 1) = x_{n+1}(t) + x_n(t),$$

by which all cells are simultaneously updated using modulo two arithmetic. The weight $\Omega(t)$ of the state of CA150 at time $t$ is defined to be the sum of all its cell variables at time $t$, or the number of dark squares in a display of the state at time $t$. Wolfram [6] gave a nice algorithm for writing down an explicit value for $\Omega(t)$.

Let $t = t_B$ denote the binary representation $t_B$ of $t$. Then, if $t_B$ involves strings of ones of length $a_k, \quad k = 1, 2, \ldots, p$, separated by strings of zeros of lengths greater than zero, these lengths being irrelevant for the algorithm, then the algorithm states that

$$\Omega(t) = \prod_{k=1}^{p} \chi_{a_k}.$$  

The numbers $\chi_n$ are shifted Jacobsthal numbers, [3], page 951, governed by

$$\chi_{n+2} = \chi_{n+1} + 2\chi_n, \quad \chi_0 = 1, \chi_1 = 3.$$  

For $t = 5$, that is $t_B = 101$ in binary, the algorithm gives $\Omega(5) = (\chi_1)2 = 9$, while for $t = 13$ or $t_B = 1101$ in binary, $\Omega(13) = \chi_1\chi_2 = 15 = \Omega(11)$, since 11 has binary representation 1011. Such results can be checked against a display of the evolution of CA150 from single seed.

In [7] a proof of the algorithm is given. Also given there is a result for the total weight $W(n)$ of the states of CA150 at the first $2^n$ time steps of its evolution from a state of single seed at $t = 0$

$$W(n) = 2^n F_{n+2},$$

involving the Fibonacci numbers. To cast (12) into a suitable form, evaluate the $\Omega(t)$ from the Wolfram algorithm, and write the answers in terms of variables $y_r = \chi_{r-1}$, noting $y_1 = 1$
and $y_2 = 3$. This leads to

\begin{align*}
W(1) &= \Omega(0) + \Omega(1) = 1 + \chi_1 = 1 + y_2 = y_12 + y_2 = 4 = 2F_3 \\
W(2) &= \Omega(0) + \Omega(1) + \Omega(2) + \Omega(3) = 1 + 2y_2 + y_3 \\
&= y_13 + 2y_1y_2 + y_3 = 12 = 4F_4 \\
W(3) &= 1 + 3y_2 + 2y_3 + y_4 + y_2^2 \\
&= y_14 + 3y_12y_2 + 2y_1y_3 + y_2^2 + y_4 = 40 = 8F_5,
\end{align*}

(13)

$W(4) = 16F_6$, and so on. The key step here is to insert factors $y_1 = 1$ into the output from the Wolfram algorithm to put the terms in each $W(n)$ into one-to-one correspondence with partitions of $n$. The results in (13) for the $W_n$ can be seen thereby to coincide with the expansions of the determinants $T_{n+1}(1,1,1,1,2) \in \mathcal{T}$. Use of Proposition 1 now allows proof for all $n$ of the result

\begin{equation}
F_{n+1} = \frac{2}{2^n} T_n(1,1,1,1,2),
\end{equation}

(14)
a determinantal identity expressing Fibonacci in terms of Jacobsthal numbers.

4. The $QR$ algorithm

Confine attention to real symmetric $n \times n$ matrices with eigenvalues

\begin{equation}
\lambda_1 > \lambda_2 \ldots \lambda_n > 0.
\end{equation}

(15)
The $QR$ factorisation of such a matrix $A$ expresses it uniquely in the form

\begin{equation}
A = QR,
\end{equation}

(16)
where $Q$ is orthogonal, and $R$ is upper triangular with positive diagonal elements. The $QR$ algorithm, based, unsurprisingly, on $QR$ factorisation, is of major importance in numerical linear algebra as a means, with widespread applications, of obtaining good approximations to all the eigenvalues of $A$. Detailed expositions can be found e.g. in [8–10]. We confine attention here to the simplest version of the algorithm, with a view entirely to exposing algebraic structure.

The $QR$ algorithm

For $r = 0, 1, 2, \ldots$, the algorithm for $A$ defines a sequence of matrices $A_r$. It does so by repetition of the following two steps

(i) factorisation of $A_r$

\begin{equation}
A_r = Q_r R_r,
\end{equation}

(17)

(ii) definition of $A_{r+1}$

\begin{equation}
R_r Q_r = A_{r+1}.
\end{equation}

(18)
The matrices $A_r$ of the sequence share with $A = A_0$ many of its key properties – symmetry, trace, determinant, eigenvalues.

The importance of the algorithm depends on its

Convergence property
\[ \lim_{r \to \infty} A_r = \Lambda = \text{diag} (\lambda_1, \lambda_2, \ldots, \lambda_n). \]  
(19)

It is proved, in contexts much more general than those to which we restrict ourselves here, in
the cited references [8–10].

By consideration of the \( n = 2 \) case, this article illustrates a study which has already been
completed [11] for real symmetric tridiagonal \( n \times n \) matrices. The purpose is to indicate the
nice algebraic structure that underlies the \( QR \) algorithm by giving closed expressions for all
the elements of all the matrices \( A_r, Q_r, R_r \): see (24–26) below. Eq. (24) yields a simple proof
of (19) in the limited context of the paper.

4.1. \( 2 \times 2 \) matrices
Consider the matrix
\[ A = A_0 = \begin{pmatrix} a & f \\ f & b \end{pmatrix}, \]  
(20)

with eigenvalues \( \lambda_+ > \lambda_- > 0 \), so that
\[ \det A = \Delta = ab - f^2 = \lambda_+ \lambda_- , \quad \text{tr} A = T = a + b = \lambda_+ + \lambda_- . \]  
(21)

Let the sequence \( S = \{ x_r, \ r \in \mathbb{Z} \} \) be governed by the difference equation
\[ x_{r+2} = Tx_{r+1} - \Delta x_r, \]  
(22)

and initial conditions \( x_0 = 1, \ x_1 = a \). This gives \( x_2 = a^2 + f^2, \ \Delta x_1 = b \), and the general
solution in the equivalent forms
\[ x_r = \alpha \lambda_+^r + \beta \lambda_-^r, \quad \alpha \beta = \frac{f^2}{(\lambda_+ - \lambda_-)^2} \]
\[ = a y_{r-1} - \Delta y_{r-2}, \quad \text{where} \quad y_r = \frac{\lambda_+^{r+1} - \lambda_-^{r+1}}{\lambda_+ - \lambda_-}. \]  
(23)

Parameterisations of all the elements of all the matrices \( A_r, Q_r, R_r \) for all \( r = 0, 1, 2, \ldots, \),
can now be presented in the form:
\[ A_r = \frac{1}{x_{2r}} \begin{pmatrix} x_{2r+1} & f \Delta^r \\ f \Delta^r & x_{2r-1} \Delta \end{pmatrix}, \]  
(24)
\[ Q_r = \frac{1}{\sqrt{x_{2r} x_{2r+2}}} \begin{pmatrix} x_{2r+1} & -f \Delta^r \\ f \Delta^r & x_{2r+1} \end{pmatrix}, \]  
(25)
\[ R_r = \frac{1}{\sqrt{x_{2r} x_{2r+2}}} \begin{pmatrix} x_{2r+2} & f T \Delta^r \\ 0 & x_{2r} \Delta \end{pmatrix}. \]  
(26)

By performing explicitly the factorisation \( A_0 = Q_0 R_0 \), it is easy to see that the expressions
(24–26) for \( r = 0 \) are correct for the given matrix \( A = A_0 \) of (20). To show that they satisfy
(17) and (18) for all \( r = 0, 1, 2, \ldots, \), requires some identities
\[ x_{r+1} x_{r-1} - x_r^2 = f 2 \Delta^{-1} \]
\[ x_{r+2} x_{r-1} - x_{r+1} x_r = f 2 T \Delta^{-1} \]
\[ x_{2r+1} x_r - x_{2r} x_{r+1} = f 2 \Delta y_{r-1} \]
\[ x_{2r} y_r - x_{2r+1} y_{r-1} = \Delta^r x_r \]
\[ x_r^2 + f 2 y_{r-1}^2 = x_{2r}. \]

An easy way to prove these identities involves insertion of explicit expressions from (23) into their left-hand sides.

Turning now to the proof that the parametrisations (24–26) satisfy (17), it is found that the \((1,1)\) and \((2,1)\) elements of the product on the right give \((A_r)_{11}\) and \((A_r)_{21}\) immediately. The \((1,2)\) element of the product must equal the \((2,1)\) element because all the \(A_r\) symmetric, and this is seen to hold upon use of (22). For the \((1,1)\) elements to agree, completing the proof of (17), requires the identity (28), with \(r\) replaced by \(2r\).

Similarly, the \((2,1)\) and \((2,2)\) elements of the product on the right side of (18) give \((A_r+1)_{21}\) and \((A_r+1)_{22}\) immediately. The \((1,2)\) element of the product must equal the \((2,1)\) element, and is found to do so, again upon use of (22). For the \((2,2)\) elements to agree, completing the proof of (18), requires the identity (28), this time with \(r\) replaced by \((2r+1)\).

Since \(\lambda_+ > \lambda_-\), it is found that
\[
\lim_{r \to \infty} x_r = \alpha \lambda_+^r. \tag{32}
\]

Recalling \(\Delta = \lambda_+ \lambda_-\), proof of (19) now follows, element by element.

4.2. Diagonalisation

Begin, following standard practice, by writing
\[
\Lambda = \begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix} = P^T A P, \tag{33}
\]
where
\[
P = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} = \frac{1}{\sqrt{\lambda_+ - \lambda_-}} \begin{pmatrix} \sqrt{a - \lambda_-} & -\sqrt{\lambda_+ - a} \\ \sqrt{\lambda_+ - a} & \sqrt{a - \lambda_-} \end{pmatrix}. \tag{34}
\]
The results \((\lambda_+ - a) > 0, \ (a - \lambda_-) > 0\) and
\[
f \cot \theta = a - \lambda_-, \tag{35}
\]
may all be proved.

Next, note that (17) and (18) imply
\[
A_{r+1} = Q_r^T A_r Q_r, \tag{36}
\]
and more generally
\[
A_r = \Omega_r^T A_0 \Omega_r, \tag{37}
\]
where

\[ \Omega_r = Q_0 Q_1 \ldots Q_{r-1}. \]  

(38)

We have already proved the key result (19). We now wish to prove that

\[ \lim_{r \to \infty} \Omega_r = P. \]  

(39)

For this purpose, it is necessary justify use of the representation

\[ \Omega_r = \frac{1}{\sqrt{x_{2r}}} \begin{pmatrix} x_r & -f y_{r-1} \\ f y_{r-1} & x_r \end{pmatrix}. \]  

(40)

Known, or easily available, results for \( \Omega_1 = Q_0 \) and \( \Omega_2 = Q_0 Q_1 \) correctly provide the cases \( r = 1, 2 \) of (40). It remains to prove that the inductive step

\[ \Omega_{r+1} = \Omega_r Q_r \]  

(41)

is true. For the \((1, 1)\) and \((2, 2)\) elements of (41) use identity (29). For the \((1, 2)\) and \((2, 1)\) elements use identity (30). To verify that (40) defines an orthogonal matrix requires (31).

Writing \( \Omega_r \) as

\[ \Omega_r = \begin{pmatrix} \cos \theta_r & -\sin \theta_r \\ \sin \theta_r & \cos \theta_r \end{pmatrix}, \]  

(42)

leads to

\[ f \cot \theta_r = \frac{x_r}{y_{r-1}} = a - \Delta \frac{y_{r-2}}{y_{r-1}}. \]  

(43)

Since \( P \) and \( \Omega_r \) define rotations of angles \( \theta \) and \( \theta_r \) respectively, it is sufficient, to complete the proof of (39), to show that \( \cot \theta_r \) approaches \( \cot \theta \) as \( r \) approaches infinity. This follows directly from the result

\[ \lim_{r \to \infty} \frac{y_{r-2}}{y_{r-1}} = \frac{1}{\lambda_r}. \]  

(44)

References

[1] Barbara R 1997 Two short proofs of Morley’s theorem Mathematical Gazette 81 447-450
[2] Scott J A 2002 Morley’s theorem once again Mathematical Gazette 86 80-81
[3] Weisstein E W 1999 CRC Concise Encyclopedia of Mathematics (Boca Raton: Chapman & Hall)
[4] Macfarlane A J (in print) The Morley tricorn: a solid figure constructed from the diagram for Morley’s theorem Mathematical Gazette
[5] Macfarlane A J 2010 Use of determinants to present identities involving Fibonacci and related numbers Fibonacci Quarterly 48 68-76
[6] Wolfram S 1983 Statistical mechanics of cellular automata Rev. Mod. Phys. 55 601-644
[7] Macfarlane A J 2004 Linear reversible second-order cellular automata and their first-order matrix equivalents J. Phys. A 37 10791-10814
[8] Olver P and Shakiban C 2006 Applied Linear Algebra (London: Pearson)
[9] Strang G 1986 Introduction to Applied Mathematics (Wellesley, MA: Wellesley Cambridge Press)
[10] Golub G H and van Loan C F 1984 Matrix Computations (Baltimore: Johns Hopkins University Press)
[11] Macfarlane A J (in preparation) The QR algorithm: explicit expressions for all the elements of all the matrices \( A_r, Q_r, R_r \) for all \( r \geq 0 \)