UNIVERSITY OF LEICESTER | SCHOOL OF BUSINESS

The Cholesky Decomposition of a Toeplitz Matrix and a Wiener-Kolmogorov Filter for Seasonal Adjustment

Emi Mise, University of Leicester
D.S.G. Pollock, University of Leicester

Working Paper No. 20 01
January 2020

THE CHOLESKY DECOMPOSITION OF A TOEPLITZ MATRIX AND A WIENER–KOLMOGOROV FILTER FOR SEASONAL ADJUSTMENT

By Emi MISE and D.S.G. POLLOCK

University of Leicester

Email: stephen_pollock@sigmapi.u-net.com
Website: http://www.le.ac.uk/users/dsgp1/

This note describes the use of the Cholesky decomposition in solving the equation $Ab = y$ when $A = A'$ is a symmetric matrix of full rank. A specialised version of the algorithm is provided for the case where A is a banded Toeplitz matrix, in which each band contains a unique repeated element and where the number of bands is considerably less than the order of the matrix, which is assumed to be large. This circumstance demands that steps should be taken to minimise the use of the computer's memory. An example is provided of the use of the algorithm in implementing a finite-sample Wiener–Kolmogorov filter aimed at removing the seasonal fluctuations from economic data.

Introduction

The Cholesky decomposition of a symmetric matrix $A = A'$ of full-rank has numerous applications. Amongst these is the solution of linear equations. If the matrix is also positive definite, then there exists a decomposition in the form of $A = LL'$, where L is a lower triangular matrix.

Such a decomposition usually requires square roots to be taken in calculating the diagonal elements of the matrix L. Since this is a time-consuming operation, it is more efficient to compute a decomposition of the form $A = LDL'$, where L is a lower triangular matrix with units on the diagonal and D is a diagonal matrix.

This decomposition is more general than the LL' decomposition, since there is no requirement that the matrix should be positive definite. If it is not, then one will find negative elements amongst the diagonal elements of D.

An important use of the Cholesky decomposition is in solving the equation $Ab = y$ when A is symmetric. A further specialisation arises when A is a banded Toeplitz matrix, which has a single repeated element along each of its diagonals.

However, in the process of calculating the decomposition, the Toeplitz structure may be overlooked, sine it cannot be exploited to obtain any extra computational efficiency. Extra efficiency and the conservation of computer memory are available whenever the matrix has a limited number of nonzero bands, whether or not they contain repeated elements.

In solving the equation $Ab = LDL'b = y$, we define $p = DL'b$. First, the equation $Lp = y$ is solved for p by a process of forward-substitution. Then, the equation $L'b = q$ is solved for b, where $q = D^{-1}p$ is formed by dividing each element of the vector p by the corresponding element of the diagonal matrix D. This equation is solved by back-substitution.

In this note, two versions of the Cholesky procedure for solving the equation $Ab = y$ will be presented. The first version makes no assumption regarding the matrix A other than its symmetry and its full rank. In the second version, A is assumed to be a narrow-band Toeplitz matrix, which may be of a considerable order. In that case, it is important to take care not to waste the storage space of the computer's memory. This requires mapping the nonzero elements of the matrix into a restricted space by altering their indices. The latter half of the paper concerns an example of the use of this version of the algorithm in effecting the seasonal adjustment of economic data.

The Standard Cholesky Decomposition

The standard LDL' decomposition can be developed with the help of an example concerning a symmetric matrix A of order 4:

$$(1)\quad \begin{aligned}\begin{bmatrix} a_{00} & a_{10} & a_{20} & a_{30} \\ a_{10} & a_{11} & a_{21} & a_{31} \\ a_{20} & a_{21} & a_{22} & a_{32} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{10} & 1 & 0 & 0 \\ l_{20} & l_{21} & 1 & 0 \\ l_{30} & l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_1 & 0 & 0 \\ 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{10} & l_{20} & l_{30} \\ 0 & 1 & l_{21} & l_{31} \\ 0 & 0 & 1 & l_{32} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} d_0 & 0 & 0 & 0 \\ d_0 l_{10} & d_1 & 0 & 0 \\ d_0 l_{20} & d_1 l_{21} & d_2 & 0 \\ d_0 l_{30} & d_1 l_{31} & d_2 l_{32} & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{10} & l_{20} & l_{30} \\ 0 & 1 & l_{21} & l_{31} \\ 0 & 0 & 1 & l_{32} \\ 0 & 0 & 0 & 1 \end{bmatrix}.\end{aligned}$$

The objective is to factorise the matrix and, in the process, to create the product

$$(2)\quad P = \begin{bmatrix} d_0 & 0 & 0 & 0 \\ l_{10} & d_1 & 0 & 0 \\ l_{20} & l_{21} & d_2 & 0 \\ 0 & l_{31} & l_{32} & d_3 \end{bmatrix} \quad : \quad \begin{bmatrix} 1 & * & * & * \\ 2 & 3 & * & * \\ 4 & 5 & 6 & * \\ 7 & 8 & 9 & 10 \end{bmatrix},$$

which contains all of the elements of the factorisation. Beside the matrix P is a matrix of numbers and asterisks that indicates a sequence in which the elements can be computed. The computations are indicated by the following list:

$$\begin{aligned}
a_{00} = d_0 &\Longrightarrow d_0 = a_{00} && \{1\} \\
a_{10} = d_0 l_{10} &\Longrightarrow l_{10} = a_{10}/d_0 && \{2\} \\
a_{11} = d_0 l_{10}^2 + d_1 &\Longrightarrow d_1 = a_{11} - d_0 l_{10}^2 && \{3\} \\
a_{20} = d_0 l_{20} &\Longrightarrow l_{20} = a_{20}/d_0 && \{4\} \\
a_{21} = d_0 l_{20} l_{10} + d_1 l_{21} &\Longrightarrow l_{21} = (a_{21} - d_0 l_{20} l_{10})/d_1 && \{5\} \\
a_{22} = d_0 l_{20}^2 + d_1 l_{21}^2 + d_2 &\Longrightarrow d_2 = a_{22} - d_0 l_{20}^2 - d_1 l_{21}^2 && \{6\} \\
a_{30} = d_0 l_{30} &\Longrightarrow l_{30} = a_{30}/d_0 && \{7\} \\
a_{31} = d_0 l_{30} l_{10} + d_1 l_{31} &\Longrightarrow l_{31} = (a_{31} - d_0 l_{30} l_{10})/d_1 && \{8\} \\
a_{32} = d_0 l_{30} l_{20} + d_1 l_{31} l_{21} + d_2 l_{32} &\Longrightarrow l_{32} = (a_{32} - d_0 l_{30} l_{20} - d_1 l_{31} l_{21})/d_2 && \{9\} \\
a_{33} = d_0 l_{30}^2 + d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 &\Longrightarrow d_3 = a_{33} - d_0 l_{30}^2 - d_1 l_{31}^2 - d_2 l_{32}^2 && \{10\}
\end{aligned}$$

These results can be generalised to the case of a matrix of an arbitrary order n. Consider a generic element of $A = [a_{ij}]$, which is on or below the diagonal of the matrix such that $i \geq j$. It is readily confirmed that

$$(4)\quad a_{ij} = \sum_{k=0}^{j} d_k l_{ik} l_{jk},$$

where d_k is the kth element of D and l_{ik} is an element from L. This equation gives rise to a generic expression for the subdiagonal elements of the ith row of L, and to an expression for the ith element of the diagonal matrix D:

(5)
$$l_{ij} = \frac{1}{d_j}\left\{a_{ij} - \sum_{k=0}^{i-1} d_k l_{ik} l_{jk}\right\},$$
$$d_i = a_{ii} - \sum_{k=0}^{j-1} d_k l_{ik}^2.$$

The solution to the equation $Ab = LDL'b = y$ is by solving the equation $Lp = y$ for p by forward substitution and, thereafter, by solving the equation $L'b = q = D^{-1}p$ by back substitution. In terms of the example, these equations are

(6)
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{10} & 1 & 0 & 0 \\ l_{20} & l_{21} & 1 & 0 \\ l_{30} & l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & l_{10} & l_{20} & l_{30} \\ 0 & 1 & l_{21} & l_{31} \\ 0 & 0 & 1 & l_{32} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} q_0 = p_0/d_0 \\ q_1 = p_1/d_1 \\ q_2 = p_2/d_2 \\ q_3 = p_3/d_3 \end{bmatrix}.$$

Their solutions are

(7)
$$\begin{aligned} p_0 &= y_0 \\ p_1 &= y_1 - l_{10}p_0 \\ p_2 &= y_2 - l_{20}p_0 - l_{21}p_1 \\ p_3 &= y_3 - l_{30}p_0 - l_{31}p_1 - l_{32}p_2 \end{aligned} \quad \text{and} \quad \begin{aligned} b_3 &= q_3 \\ b_2 &= q_2 - l_{32}b_3 \\ b_1 &= q_1 - l_{31}b_3 - l_{21}b_2 \\ b_0 &= q_0 - l_{30}b_3 - l_{20}b_1 - l_{10}b_1. \end{aligned}$$

The following *Pascal* procedure uses the above equations in the process of factorising $A = LDL'$ and of solving the equation $Ab = LDL'b = y$. The complete matrix A is passed to the procedure. It is returned with the subdiagonal elements of L replacing its own subdiagonal elements, and with the elements of D along its principal diagonal. From the returned matrix, it is easy to calculate the determinant of the original matrix A by forming the product of the elements of D. The vector y, which is passed to the procedure, becomes, in succession, the vectors p, q and b, all of which occupy the space of y.

(8)
```
procedure Cholesky(n : integer;
                   var a : matrix;
                   var y : vector);

  var
    i, j, k : integer;

begin

  for i := 0 to n do
    for j := 0 to i do
      begin {i, j}
        for k := 0 to j - 1 do
          a[i, j] := a[i, j] - a[k, k] * a[i, k] * a[j, k];
        if j < i then
          begin
```

```
            a[i, j] := a[i, j]/a[j, j];
            a[j, i] := 0.0;
          end;
        end; {i, j}

    {Forward Solve}
      for i := 0 to n do
        for j := 0 to i - 1 do
          y[i] := y[i] - a[i, j] * y[j];

    {Divide by the Diagonal}
      for i := 0 to n do
        y[i] := y[i] - a[i, j]/a[i, i]

    {Back Solve}
      for i := n downto 0 do
        for j := n downto i + 1 do
          y[i] := y[i] - a[j, i] * y[j];

    end; {Cholesky}
```

The Decomposition of a Banded Toeplitz Matrix

The objective is to create an algorithm for decomposing a symmetric narrow banded Toeplitz matrix. Only the lower triangular nature of the factor L and the limitation in the number of bands need to be taken into account. It helps if, in the first instance, the fact is ignored that each of the bands contains only a single repeated value. The efficiency of the algorithm derives from its conservation of storage space and its avoidance of calculations that produce zero-valued elements.

The potential for conserving memory space can be demonstrated by the case of a matrix factor that has $n = 5$ rows and only $q = 2$ subdiagonal bands. The reformatted matrix that eliminates the majority of the zero-valued elements of L will be denoted by M. The rearrangement the elements in order to conserve storage is indicated as follows:

$$(9) \qquad L = \begin{bmatrix} l_{00} & 0 & 0 & 0 & 0 \\ l_{10} & l_{11} & 0 & 0 & 0 \\ l_{20} & l_{21} & l_{22} & 0 & 0 \\ 0 & l_{31} & l_{32} & l_{33} & 0 \\ 0 & 0 & l_{42} & l_{43} & l_{44} \end{bmatrix} \longrightarrow M = \begin{bmatrix} l_{00} & 0 & 0 \\ l_{11} & l_{10} & 0 \\ l_{22} & l_{21} & l_{20} \\ l_{33} & l_{32} & l_{31} \\ l_{44} & l_{43} & l_{42} \end{bmatrix} = \begin{bmatrix} m_{00} & 0 & 0 \\ m_{10} & m_{11} & 0 \\ m_{20} & m_{21} & m_{22} \\ m_{30} & m_{31} & m_{32} \\ m_{40} & m_{41} & m_{42} \end{bmatrix}$$

It will be observed that the ordering of the nonzero elements in each of the rows is reversed in passing from L to M and that most of the zero-valued elements are eliminated. The nonzero element l_{ij} within L becomes the element $m_{i,i-j} = l_{i,j}$ within M. Conversely, $m_{i,j} = l_{i,i-j}$.

The decomposition of the symmetric matrix $A = [\alpha_{i,i-j}]$ can be represented as follows:

$$\begin{bmatrix} \alpha_{00} & \alpha_{11} & \alpha_{22} & \{\alpha_{33}\} \\ \alpha_{11} & \alpha_{10} & \alpha_{21} & \alpha_{32} \\ \alpha_{22} & \alpha_{21} & \alpha_{20} & \alpha_{31} \\ \{\alpha_{33}\} & \alpha_{32} & \alpha_{31} & \alpha_{30} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{11} & 1 & 0 & 0 \\ m_{22} & m_{21} & 1 & 0 \\ \{m_{33}\} & m_{32} & m_{31} & 1 \end{bmatrix} \begin{bmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_1 & 0 & 0 \\ 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & m_{11} & m_{22} & \{m_{33}\} \\ 0 & 1 & m_{21} & m_{32} \\ 0 & 0 & 1 & m_{31} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d_0 & 0 & 0 & 0 \\ d_0 m_{11} & d_1 & 0 & 0 \\ d_0 m_{22} & d_1 m_{21} & d_2 & 0 \\ \{d_0 m_{33}\} & d_1 m_{32} & d_2 m_{31} & d_3 \end{bmatrix} \begin{bmatrix} 1 & m_{11} & m_{22} & \{m_{33}\} \\ 0 & 1 & m_{21} & m_{32} \\ 0 & 0 & 1 & m_{31} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The change of notation whereby $a_{ij} = \alpha_{i,i-j}$ has no effect within the context of the matrix A, since it does not change the values of the elements. However, the mappping $a_{i,i-j} = \alpha_{ij} \longrightarrow m_{ij}$, which occurs at the start of the computations, is crucial for the efficient storage of the nonzero elements of the lower triangle of the symmetric matrix. Thus, at the outset, the nonzero elements of the lower triangle of A are stored in the matrix M which will eventually contain all the elements of the decomposition. There are braces surrounding the elements α_{33} and m_{33} as a reminder that these may be zero-valued, in which case, there will be $q = 2$ supradiagonal and subdiagonal bands in the matrix A and q subdiagonal bands in its factor $L = [m_{i,i-j}]$.

The computations are indicated by the following list:

$$\begin{aligned}
\alpha_{00} = d_0 &\Longrightarrow d_0 = \alpha_{00} \\
\alpha_{11} = d_0 m_{11} &\Longrightarrow m_{11} = \alpha_{11}/d_0 \\
\alpha_{10} = d_0 m_{11}^2 + d_1 &\Longrightarrow d_1 = \alpha_{10} - d_0 m_{11}^2 \\
\alpha_{22} = d_0 m_{22} &\Longrightarrow m_{22} = \alpha_{22}/d_0 \\
\alpha_{21} = d_0 m_{22} m_{11} + d_1 m_{21} &\Longrightarrow m_{21} = (\alpha_{21} - d_0 m_{22} m_{11})/d_1 \\
\alpha_{20} = d_0 m_{22}^2 + d_1 m_{21}^2 + d_2 &\Longrightarrow d_2 = \alpha_{20} - d_0 m_{22}^2 - d_1 m_{21}^2 \\
\{\alpha_{33} = d_0 m_{33} &\Longrightarrow m_{33} = \alpha_{33}/d_0\} \\
\alpha_{32} = \{d_0 m_{33} m_{11}\} + d_1 m_{32} &\Longrightarrow m_{32} = (\alpha_{32} - \{d_0 m_{33} m_{11})/d_1\} \\
\alpha_{31} = \{d_0 m_{33} m_{22}\} + d_1 m_{32} m_{21} + d_2 m_{31} &\Longrightarrow m_{31} = (\alpha_{31} - \{d_0 m_{33} m_{22}\} - d_1 m_{32} m_{21})/d_2 \\
\alpha_{30} = \{d_0 m_{33}^2\} + d_1 m_{32}^2 + d_2 m_{31}^2 + d_3 &\Longrightarrow d_3 = \alpha_{30} - \{d_0 m_{33}^2\} - d_1 m_{32}^2 - d_2 m_{31}^2
\end{aligned} \tag{11}$$

These results can be generalised to the case of a matrix of an arbitrary order n. Consider a generic element of $A = [\alpha_{ij}]$ which is on or below the diagonal of the matrix such that $i \geq j$. It is readily confirmed that

$$\alpha_{ij} = \sum_{k=r}^{p} d_k m_{i,i-k} m_{p,p-k}, \quad p = i - j, \quad r = \text{Max}(0, i - q), \tag{12}$$

where m_{ik} is an element from M and $d_k = m_{k0}$ is the kth element of D. This equation gives rise to a generic expression for the subdiagonal elements of the jth column of M, and to an expression for the ith element of the diagonal matrix D:

$$\begin{aligned}
m_{ij} &= \frac{1}{d_j}\left\{\alpha_{ij} - \sum_{k=r}^{p-1} d_k m_{i,i-k} m_{p,p-k}\right\}, \\
d_i &= \alpha_{i0} - \sum_{k=r}^{i-1} d_k m_{i,i-k}^2.
\end{aligned} \tag{13}$$

In comparing these expressions with those under (4), one should note that the index k is now associated with a negative sign, which is in consequence of the reversal of the order of the elements within the matrix M. The initial value of the index is $r \geq 0$, which differs from 0 when the number of subdiagonal bands is $q < n$, where n is the limiting index. The case were A is a Toeplitz matrix of order $n+1$ and with q supradiagonal and q subdiagonal bands is of primary concern. For the example above, where $n = 3$ and and $q = 2$, the matrix in question is

(14) $$A = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & 0 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_0 & \gamma_1 & \gamma_0 & \gamma_1 \\ 0 & \gamma_2 & \gamma_1 & \gamma_0 \end{bmatrix},$$

which can be stored as the vector $[\gamma_0, \gamma_1, \gamma_2]$. These elements are transferred to the matrix M as the calculations proceed by setting $m_{ij} = \gamma_j$. Then, the elements m_{ij} will serve in place of the elements α_{ij} in the computations that are represented by the display under (11) and, eventually, they will become the final products that are to be found in the matrix M

The solution to the equation $Ab = LDL'b = y$ is via the following equations

(15)
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{11} & 1 & 0 & 0 \\ m_{22} & m_{21} & 1 & 0 \\ \{m_{33}\} & m_{32} & m_{31} & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & m_{11} & m_{22} & \{m_{33}\} \\ 0 & 1 & m_{21} & m_{32} \\ 0 & 0 & 1 & m_{31} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} q_0 = p_0/d_0 \\ q_1 = p_1/d_1 \\ q_2 = p_2/d_2 \\ q_3 = p_3/d_3 \end{bmatrix}.$$

Their solutions are

(16) $$\begin{aligned} p_0 &= y_0 \\ p_1 &= y_1 - m_{11}p_0 \\ p_2 &= y_2 - m_{22}p_0 - m_{21}p_1 \\ p_3 &= y_3 - \{m_{33}p_0\} - m_{32}p_1 - m_{31}p_2 \end{aligned} \quad \text{and} \quad \begin{aligned} b_3 &= q_3 \\ b_2 &= q_2 - m_{31}b_3 \\ b_1 &= q_1 - m_{32}b_3 - m_{21}b_2 \\ b_0 &= q_0 - \{m_{33}b_3\} - m_{22}b_2 - m_{11}b_1. \end{aligned}$$

The braces surrounding the element α_{33} continue to serve as a reminder that this may be zero-valued.

The following procedure implements the algorithm that solves the equation $Ab = y$ in the case where A is a full-rank symmetric Toeplitz matrix with q subdiagonal and supradiagdonal bands.

(17)
```
procedure Toeplitz(var n, q : integer;
                   var gamma : vector;
                   var y : longVector);

  var
    i, j, k, p, r, s : integer;
    m : array [0..maxArray, 0..maxOrder] of real; {longMatrix}

 begin

{Factorise}
  for i := 0 to n do
    begin {i}
      r := Max(0, i - q);
      s := Min(i, q);
      for j := s downto 0 do
        begin {j}
```

```
            p := i − j;
            m[i, j] := gamma[j];
             for k := r to p − 1 do
               m[i, j] := m[i, j] − m[k, 0] ∗ m[i, i − k] ∗ m[p, p − k];
             if j > 0 then
               m[i, j] := m[i, j]/m[p, 0];
          end;{j}
      end;{i}

  {Forward Solve}
    for i := 0 to n do
      for j := Max(0, i − q) to i − 1 do
        y[i] := y[i] − m[i, i − j] ∗ y[j];

  {Divide by the diagonal}
    for i := 0 to n do
      y[i] := y[i]/m[i, 0];

  {Back Solve}
    for i := n downto 0 do
      for j := Min(i + q, n) downto i + 1 do
        y[i] := y[i] − m[j, j − i] ∗ y[j];

  end;{Toeplitz}
```

In this procedure, various restrictions are imposed on the range of j in consequence of the limitation of the number q of supradiagonsl and subdiagonal bands in the matrix A. The first of these restrictions applies directly to the matrix M. The number of nonzero elements in its ith row is restricted by $s = Min(i, q)$, which implies that the row contains the full number of nonzero elements only when $i \geq q$.

Next is the restriction $r = Max(0, i - q)$ on the initial value of the index k. With the index k running from $r = i - q$ to $p = i - j$, this corresponds to the fact that α_{ij} of (10) comprises a full set of $q + 1$ elements only when $j = 0$, which is when it is a diagonal element of A.

The remaining restrictions affect the processes of forward substitution and back substitution. First, there is the restriction that sets the initial value of j within the forwards process to $Max(0, i-q)$. This is entails the restriction on $m[i, i-j]$ that $i-j \leq q$; and it accommodates the case where $i-q \leq 0$.

Then, within the backwards process, there is the restriction that the initial value of the index of j is $Min(i + q, n)$. This is entails the restriction on $m[j, j - i]$ that $j - i \leq q$; and it accommodates the case where the index i has declined by less than q.

A Wiener–Kolmogorov Filter for Eliminating Seasonal Fluctuations

The procedure *Toeplitz* can be illustrated by considering a Wiener–Kolmogorov filter for eliminating the seasonal fluctuations from an economic data sequence. The business of seasonal adjustment is commonly subsumed under the analysis of unobserved components, whereby a economic data sequence is expressed as a sum of separate components that are assumed to be statistically independent.

A taxonomy of unobserved components has been described in relation to the *STAMP* computer program by Koopmans *et al.* (1995). A taxonomy that is appropriate to the purposes of this paper expresses the logarithmic data sequence $d(t) = \{d_t = \ln(D_t); t = 0, \pm 1, \pm 2, \ldots\}$ as

$$d(t) = \pi(t) + \zeta(t) + \xi(t) + \eta(t). \tag{18}$$

Here, $\pi(t)$ is a trend function that is liable to be a polynomial functions of the index t, albeit that the trend might be a more complicated function that is designed to absorb major structural breaks in the data. (See, for example, Pollock 2016.) The remaining components, which are sequences of zero mean, are $\zeta(t)$, which represents a secular cycle or business cycle, $\xi(t)$, which represents the seasonal fluctuations, and $\eta(t)$, which represents an irregular component.

A common recourse it to merge first two components of (18) to form a trend-cycle function $\tau(t) = \pi(t) + \zeta(t)$, which is liable to be described by an integrated moving average with an autoregressive component consisting of a twofold difference operator. An alternative recourse is to merge the secular cycle $\zeta(t)$ with the irregular component $\eta(t)$ to create a component $\upsilon(t) = \zeta(t) + \eta(t)$ that could be modelled via a stationary ARMA process. This is in line with the procedures of this paper.

The immediate purpose is to derive a filter that can be relied on to remove the seasonal fluctuations from the data sequence $y(t) = \xi(t) + \upsilon(t)$ that has been reduced to stationarity by the removal of the trend component $\pi(t)$. There is no advantage then in representing $\upsilon(t)$ by an elaborate ARMA process; and, for the purpose of deriving the filter from a statistical criterion, we shall describe $\upsilon(t) = \eta(t)$ as a white-noise process.

The resulting heuristic model of the detrended data may be represented in a z-transform notation by

$$\begin{aligned} y(z) &= \xi(z) + \eta(z) \\ &= \frac{P(z)}{\Sigma(z)}\nu(z) + \eta(z). \end{aligned} \tag{19}$$

Here, $y(z) = \sum_t y_t z^t$ represents the z-transform of a sequence that is commonly assumed to be doubly infinite. However, the convergence of z-transform depends on the assumption that the sequence $\{y_t; t = 0, \pm 1, \pm 2, \ldots\}$ is absolutely summable. The same consideration applies the sequences $\{\nu_t; t = 0, \pm 1, \pm 2, \ldots\}$ and $\{\eta_t; t = 0, \pm 1, \pm 2, \ldots\}$ that are, otherwise, assumed to be generated by white-noise process with the variances $V(\nu_t) = \sigma_\nu^2$ and $V(\eta_t) = \sigma_\eta^2$, respectively.

The sequence $\{\nu_t; t = 0, \pm 1, \pm 2, \ldots\}$ is the forcing function of the seasonal process. It is subjected to a transfer function represented by the rational function $P(z)/\Sigma(z)$. The denominator of the function is the polynomial

$$\Sigma(z) = \frac{1 - z^s}{1 - z} = 1 + z + z^2 + \cdots + z^{s-1}, \tag{20}$$

wheras the numerator is

$$P(z) = \Sigma(\rho z) = \frac{1 - \rho^s z^s}{1 - \rho z} = 1 + \rho z + \rho^2 z^2 + \cdots + \rho^{s-1} z^{s-1}. \tag{21}$$

The poles of the denominator of the transfer function are the complex numbers $\exp(\mathrm{i}2\pi j/s); j = 1, \ldots, s-1$, which lie on the circumference of the unit circle in the complex plane at angles that correspond to the seasonal frequency and its harmonics. These are responsible for driving the seasonal fluctuations. The zeros of numerator are the values $\rho\exp(\mathrm{i}2\pi j/s); j = 1, \ldots, s-1$. They serve to confine the fluctuations to narrows bands of frequencies that surround the seasonal frequencies.

The presence of complex roots of unit modulus within the polynomial $\Sigma(z)$ implies that the process generating $\{y_t; t = 0, \pm1, \pm2, \ldots\}$ is nonstationary in amplitude. It may be reduced to stationarity by multiplying throughout by the denominator of the filter to give

$$\begin{aligned}\Sigma(z)y(z) &= P(z)\nu(z) + \Sigma(z)\eta(z)\\ &= \delta(z) + \kappa(z) = g(z).\end{aligned} \tag{22}$$

The z-transform of the Wiener–Kolmogorov filter that serves, equally, to extract $\eta(z)$ from $y(z)$ and $\Sigma(z)\eta(z)$ from $\Sigma(z)y(z)$ is

$$\beta(z) = \frac{\sigma_\eta^2\Sigma(z^{-1})\Sigma(z)}{\sigma_\eta^2\Sigma(z^{-1})\Sigma(z) + \sigma_\nu^2 P(z^{-1})P(z)} = \frac{\Sigma(z^{-1})\Sigma(z)}{\Pi(z^{-1})\Pi(z)}, \tag{23}$$

where

$$\Pi(z^{-1})\Pi(z) = \Sigma(z^{-1})\Sigma(z) + \lambda P(z^{-1})P(z), \quad \text{with} \quad \lambda = \frac{\sigma_\nu^2}{\sigma_\eta^2}. \tag{24}$$

.

The inclusion of the term $\sigma_\eta^2\Sigma(z^{-1})\Sigma(z)$ in the denominator of (23) means that the gain of the Wiener–Kolmogorov filter cannot exceed unity. However, to ensue that unity is attained at the zero frequency, it is appropriate to normalise the filter by dividing the coefficients of the numerator and denominator by their respective sums.

The effects of a seasonal adjustment filter are best represented by its frequency-response function, which shows how the filter alters the amplitudes of the sinusoidal elements of which a stationary data sequence is composed. Figure 1 shows the frequency response function of two such filters, wherein the parameter values are $\rho = 0.99$ and $\lambda = 0.5$, which give rise to a frequency response function with narrow clefts at the seasonal frequencies, and $\rho = 0.8$ and $\lambda = 0.5$, which give rise to one with wide clefts.

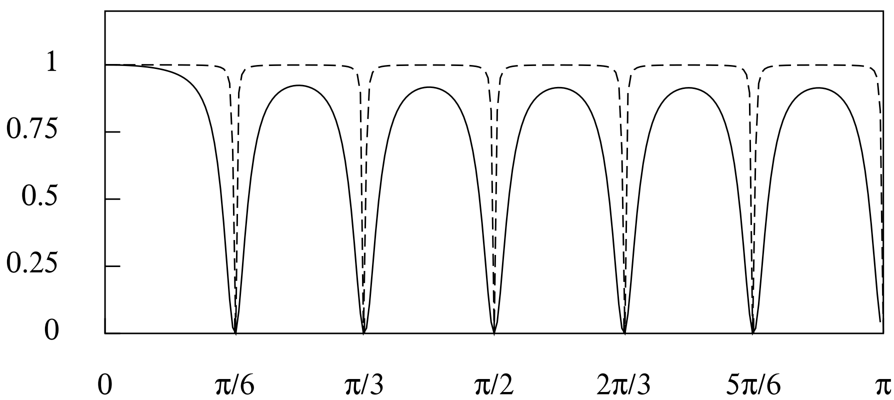

Figure 1. The frequency response functions of the ordinary seasonal adjustment filter for monthly data with $\lambda = 0.5.$ and $\rho = 0.8$ (the solid line) and with $\lambda = 0.5.$ and $\rho = 0.99$ (the dashed line).

The Finite-Sample Wiener–Kolmogorov Filter

To derive the finite-sample version of the Wiener–Kolmogorov filter, consider a vector $y = [y_0, y_1, \ldots, y_{T-1}]'$ of T values drawn from the process represented by $y(z)$. In accordance with equation (19), the vector may be decomposed as

$$y = \xi + \eta. \tag{25}$$

To create a vector of a stable amplitude, the vector y must be transformed by a matrix $\Sigma_s = \Sigma(L_T)$ of order T, which is the finite-sample analogue of the operator $\Sigma(z)$. This matrix is derived by replacing the argument z by the matrix lag operator $L_T = [e_1, \ldots, e_{T-1}, 0]$ of order T, which is derived from the identity matrix $I_T = [e_0, e_1, \ldots, e_{T-1}]'$ by deleting the leading column and by adding a column of zeros to the end of the array.

When applying the matrix operator to the vector y, the first s elements of the product, which are in g_*, are liable to be discarded:

$$\Sigma(L_T)y = \begin{bmatrix} S'_* \\ S' \end{bmatrix} y = \begin{bmatrix} g_* \\ g \end{bmatrix}. \tag{26}$$

Here S', is a matrix of order $(T - s + 1) \times T$, of which the jth row contains the s unit coefficients of $S(z)$ preceeded by $j - 1$ zeros and followed by zeros. The matrix S_* of order $(s - 1) \times T$ contains a leading lower-triangular matrix filled with units and a following matrix of order $(s - 1) \times (T - s + 1)$ full of zeros.

In common with $\Sigma(L_T)$, the finite-sample analogue of the operator $P(z)$ has a Toeplitz structure as follows:

$$P(L_T) = \left[\begin{array}{ccccccccc} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \rho & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \rho^{s-2} & \rho^{s-3} & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\ \hline \rho^{s-1} & \rho^{s-2} & \cdots & \rho & 1 & 0 & \cdots & 0 & 0 \\ 0 & \rho^{s-1} & \cdots & \rho^2 & \rho & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho^{s-1} & \rho^{s-2} & \rho^{s-3} & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & \rho^{s-1} & \rho^{s-2} & \cdots & \rho & 1 \end{array}\right] = \begin{bmatrix} R'_* \\ R' \end{bmatrix}. \tag{27}$$

Thus, $P(L_T)$ becomes $\Sigma(L_T)$ when $\rho = 1$.

Applying S' to the equation $y = \xi + \eta$, representing the seasonally fluctuating data, gives

$$\begin{aligned} S'y &= R'\nu + S'\eta \\ &= \delta + \kappa = g. \end{aligned} \tag{28}$$

This is just a segment of $T - s$ elements drawn from the process represented by equation (22).

The expectations and the dispersion matrices of the component vectors of g are

$$\begin{aligned} E(\delta) &= 0, \qquad D(\delta) = \sigma_\nu^2 R'R, \\ E(\kappa) &= 0, \qquad D(\kappa) = \sigma_\eta^2 S'S. \end{aligned} \tag{29}$$

The difficulty of estimating the vector $\xi = y - \eta$ of seasonal fluctuations directly is that some starting values or initial conditions are required in order to define the value at time $t = 0$. However, since η is from a stationary mean-zero process, it requires only zero-valued initial conditions. Therefore, the starting-value problem can be circumvented by concentrating on the estimation of η, wherafter an estimate of $\xi = y - \eta$ is readily available. The estimates of ξ and η will be denoted by the roman letters x and h respectively.

The conditional expectation of η, given the transformed data $g = S'y$, is provided by the formula

$$\begin{aligned} h &= E(\eta|g) = E(\eta) + C(\eta, g)D^{-1}(g)\{g - E(g)\} \\ &= C(\eta, g)D^{-1}(g)g, \end{aligned} \tag{30}$$

where the second equality follows in view of the zero-valued expectations of η and g. Within this expression, there are

$$D(g) = \sigma_\nu^2 R'R + \sigma_\eta^2 S'S \quad \text{and} \quad C(\eta, g) = \sigma_\eta^2 S. \tag{31}$$

Putting these details into (30) gives the following estimate of η:

$$\begin{aligned} h &= \sigma_\eta^2 S(\sigma_\nu^2 R'R + \sigma_\eta^2 S'S)^{-1}S'y \\ &= S(S'S + \lambda R'R)^{-1}S'y, \end{aligned} \tag{32}$$

whence

$$\begin{aligned} x = E(\xi|g) = y - E(\eta|g) &= y - h \\ &= \{I - S(S'S + \lambda R'R)^{-1}S'\}y. \end{aligned} \tag{33}$$

A simple procedure for calculating h begins by solving the equation

$$(S'S + \lambda R'R)b = S'y = g \tag{34}$$

for the value of b. Thereafter, one can generate $h = Sb$.

The matrix $S'S + \lambda R'R'$ is a narrow-band Toeplitz matrix with the structure of the variance-covariance matrix of a moving-average process of order $s - 1$. The matrix can be encoded in a vector of s elements. The solution to equation (34) may be found via a Cholesky factorisation that sets $S'S + \lambda R'R = LDL'$, where L is a lower-triangular matrix with a limited number of nonzero bands and D is a diagonal matrix. The system $LDL'b = g$ may be cast in the form of $Lp = g$ and solved for p. Then, $L'b = D^{-1}p$ can be solved for b whence $h = Sb$ can be derived.

It will be observed that the equation (32) may also be written as

$$h = (SL'^{-1})D^{-1}(L^{-1}S')y, \tag{35}$$

where SL'^{-1} is an upper-triangular matrix, and where its transpose $L^{-1}S'$ is a lower-triangular matrix. This equation corresponds to a method of bi-directional filtering in which $p = L^{-1}S'y$ represents a real-time filtering and $h = SL'^{-1}D^{-1}p$ represent a reverse-time filtering, which is also described as a smoothing operation.

Additional Procedures

The facilities for calculating the seasonally-adjusted data sequence in h must include procedures for calculating $S'y = g$ and $h = Sb$. The first of these procedures is listed as follows:

(36)
```
procedure SprimeY(q, n : integer;
                  var sigma : vector;
                  var y : longVector);
```

```
  var
    t, j : integer;
    store : real;

begin
  for t := 0 to n - q do
    begin {t}
      store := 0.0;
      for j := 0 to q do
        store := store + sigma[q - j] * y[t + j];
      y[t] := store;
  end;{t}
end;{SPrimeY}
```

This procedure has a degree of generality that allows the array *sigma* to contain something other than the s units that are the coefficients of the polynomial $\Sigma(z)$ of (20), or the s coefficients of $P(z)$. Also, the integer q, which is the maximum index of *sigma*, may differ from $s-1$, which is the degree of $\Sigma(z)$ and $P(z)$. The procedure is readily intelligible as an implementation of the formula

$$g_t = \sum_{j=0}^{q} \sigma_{q-j} y_{t+j}; \quad q = s - 1, \tag{37}$$

which corresponds to the multiplications in the lower part of (26), wherein matrix S' has the same structure as R' of (27), but with $\rho = 1$. Because the nonzero elements of S' lie on and above the principal diagonal of the matrix, the elements of $g = S'y$ must be calculated by running from top to bottom of the vector g.

The second procedure is marginally more complicated in consequence of the fact that there are end-effects to contend with in the matrices S and R. That is to say, the leading and trailing rows of the matrices S and R comprise fewer than the full set of s (or $q+1$) coefficients:

(38)

```
procedure Sy(q, n : integer;
              var sigma : vector;
              var y : longVector);
  var
    t, j, r, k : integer;
    store : real;

begin
  for t := n downto 0 do
    begin
      r := Max(0, q - t);
      k := Min(q, n - t);
      store := 0.0;
      for j := r to k do
        store := store + sigma[j] * y[t - q + j];
      y[t] := store;
  end;
end;{Sy}
```

The procedure implements the formula

$$h_t = \sum_{j=r}^{k} \sigma_j y_{t-q+j}; \quad q = s - 1, \tag{37}$$

where, in order to accommodate the end-effects, the limits of the summation are $r := Max(0, q - t)$ and $k := Min(q, n - t)$. Given that the nonzero elements of S lie on and above the principal diagonal of the matrix, the elements of $h = Sd$ must be calculated by running from bottom to top of the vector h.

Seasonal Adjustment in Practice

Central statistical offices use two varieties of methods for seasonal adjustment. In the past, the dominant methods have been the venerable *X-11* procedure of Shiskin, Young and Musgrave (1967) and its derivatives. The *X-11* program has been fully documented in a monograph of Ladiray and Quenneville (2001).

Recently, model-based methods have gained favour. These are representes, primarily, by the highly competent *TRAMO–SEATS* program of Augustin Maravall—see Gomez and Maravall (1997) and Caporello and Maravall (2004). This program follows the prescriptions of Hillmer and Tao (1982) regarding the canonical decomposition of time series affected by seasonal and cyclical variations.

Other model-base methods of seasonal adjustment are offered by the *Captain Toolbox*, described, originally, by Young, Pedregal, and Tych (1999) and, more recently, by Taylor (2017), and by the *STAMP* program of Koopmans, Harvey, Doornik, and Shephard (1995). A broad perspective on model-based business-cycle analysis and seasonal adjustment has been provided by Kaiser and Maravall (2001). It is with the model-based methods that the methods of this paper can be compared most directly.

The model-based methods are the products of a dominant opinion amongst economists that economic investigations should be conducted within the context of well-defined models of economic activities. However, there can be some advantages in exercising direct control over the parameters that determine the characteristics of the seasonal-adjustment filter. Then, the parameters may be specified in the light of the salient spectral characteristics of the data. Moreover, there may be difficulties in estimating a model in consequence of the heterogeneous nature of the data,

The methods that are presented in this paper do not require the estimation any model. The appropriate values of parameters λ and ρ, which determine the width of the clefts in the frequency response of the filter, can be determined via an appraisal of the the periodogram of the detrended data sequence.

In practice, having set λ to an arbitrary value (0.5 works well), one can rely upon the value of ρ to determine the appropriate filter. It is notable that the frequency response function of Figure 1 with $\lambda = 0.5$ and $\rho = 0.8$ is indistinguishable from the analogous response function estimated by *TRAMO–SEATS* by applying the airline passenger model of Box and Jenkins (1976) to the logarithms of the monthly international airline passenger totals from January 1949 to December 1960. This is shown in Figure 2.

Another problem that affects the time-domain methods of seasonal adjustment is that they nullify completely only the elements at the seasonal seasonal frequency and its harmonics. The seasonal fluctuations may comprise elements at adjacent frequencies that also need to be removed from the data. A testimony to this problem has been provided by McElroy and Roy (2017), who have provided a means of detecting residual seasonal effects in seasonally-adjusted date. The issue has also been addressed by Findley *et al.* (2005).

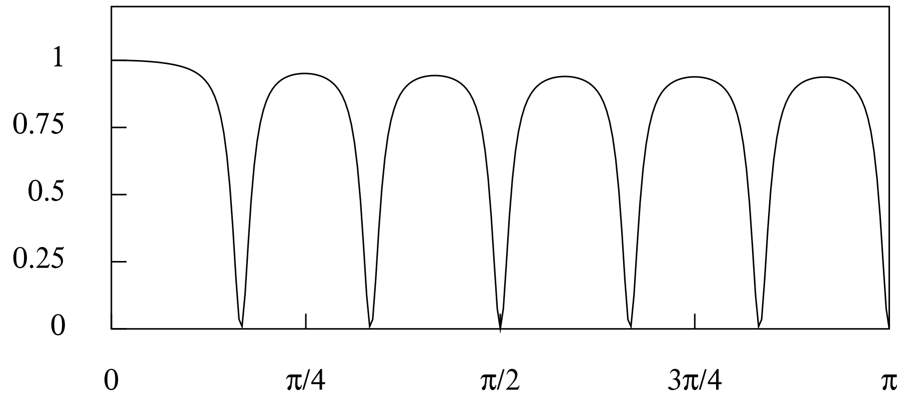

Figure 2. The frequency response of the seasonal-adjustment filter associated with the monthly airline passenger model.

The problem can arise in consequence of a variety of data anomalies that affect the regularity of the seasonal fluctuations. These include calendar effects, holidays, strikes and other untoward events. Methods of dealing with such irregularities by adjusting the data directly have been described, recently, by Attal-Toubert *et al.* (2018) and by Ladiray (2018).

One way of eliminating a wider band of elements in the vicinities of the seasonal frequencies, which can be based on the current procedure, is to create offset filters that are targeted at the frequencies on either side of the seasonal frequencies. Then, such filters can be applied in series with a central filter targeted at the seasonal frequencies. The program *SEADOS,* which is associated with this paper, contains such a facility.

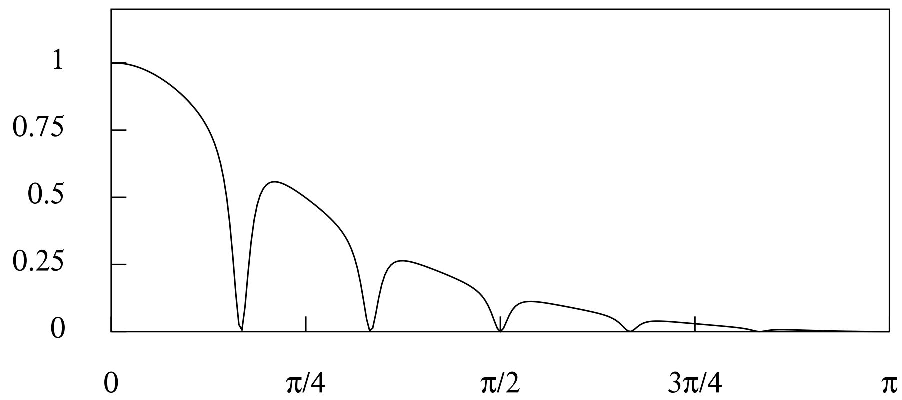

Figure 3. The frequency response of the trend extraction filter associated with the monthly airline passenger model.

Estimating a Trend-Cyle Function

The procedure of Hillmer and Tao (1982), which is adopted by the *TRAMO–SEATS* program, decomposes the data into a trend-cycle component, a seasonal component and an irregular component. The seasonally adjusted data is the sum of the trend-cycle component and the irregular component. The trend-cycle component on its own is also of primary interest. Figure 3 shows the frequency response function of the trend-cycle extraction filter associated with the the airline passenger model of Box and Jenkins (1976) that is employed by *TRAMO–SEATS.*

In the *SEADOS* program, the fundamental trend is liable to be represented by a polynomial function that is fitted to the data by a least-squares or a weighted least-squares regression. A trend-cycle function, which is equivalent to the function derived from the airline passenger model, may be created by applying a smoothing filter to residuals from the polynomial regression that have been seasonally adjusted. The smoothed sequence is then added back to the polynomial trend to create the trend-cycle.

The unidirectional smoothing filter $M(z)$ is a simple second-order moving average incorporating a zero at the limiting Nyquist frequency of π, with a value of -1, and a second zero $-1/\kappa \in (0, -1)$:

$$M(z) = \frac{(1+z)(1+\kappa z)}{2(1+\kappa)} = \frac{1+(1+\kappa)z+\kappa z^2}{2(1+\kappa)}, \quad \text{with} \quad \kappa \in [0,1], \tag{38}$$

The bidirectional filter $M(z^{-1})M(z)$ is applied to the seasonally-adjusted data sequence using the procedures *SPrimey* and *Sy* of (36) and (38) in sequence, with the array *sigma* holding the coefficients of $M(z)$. A greater attenuation of the high-frequency elements is achieved using the twofold $M^2(z)$ filter.

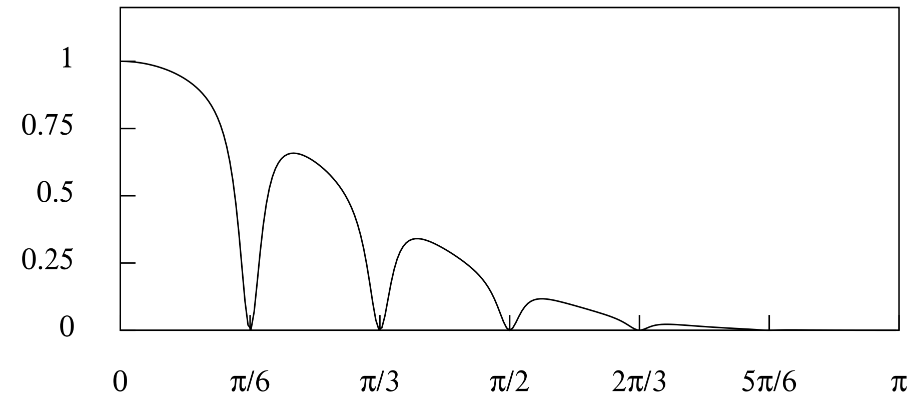

Figure 4. The frequency response of the $M^2(z)$ trend-cycle extraction filter that mimicks that of the monthly airline passenger model.

Figure 4 shows the frequency response function of a filter that compounds the bidirectionsl $M^2(z)$ smoothing filter with seasonal-adjustment filter of which the frequency response is depicted in Figure 1 by the unbroken line. In this case, the smoothing parameter is $\kappa = 0.4$. The effect of applying this combined filter to the logarithms of the monthly index of U.S. total sales from January 1953 to December 1964. is shown in Figure 5.

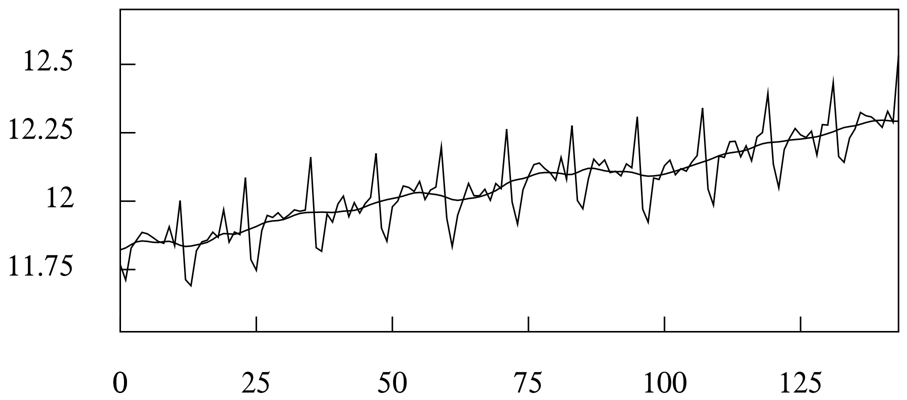

Figure 5. The logarithms of U.S. total retail sales from January 1953 to December 1964 with an interpolated trend-cycle function.

The *SEADOS* Program

A program that implements the procedures that have been described in this paper is available at the following address:

`http://www.le.ac.uk/users/dsgp1/`

It is to be found under the legend *SEADOS: A Program For Seasonal Adjustment in the Time Domain.* The *Pascal* code of the program is also provided at this address. The program has been compiled using the *Free Pascal* compiler, which is freely available on the web.

Embedded in the program are brief descriptions of its facilities and it its functions. These should provide sufficient guidance for operating the program.

References

Attal-Toubert, K., D. Ladiray, M. Marini and J. Palate, (2018), Moving Trading-Day Effects with X-13 ARIMA-SEATS and TRAMO-SEATS, Chapter 6 in Mazzi, G.L., D. Ladiray, and D.A. Rieser, (eds.), *Handbook on Seasonal Adjustment, 2018 edition,* Eurostat: Publications Office of the European Union, Luxembourg.

Box, G.E.P., and G.M. Jenkins, (1976), *Time Series Analysis: Forecasting and Control, Revised Edition,* Holden Day, San Francisco.

Caporello G., and A. Maravall, (2004), *Program TSW, Revised Reference Manual,* Servicio de Estudios, Banco de España.

Findley, D.F., T.S. McElroy and K.C. Wills, (2005), Modifications of SEATS' Diagnostic for Detecting Over- or Underestimation of Seasonal Adjustment Decomposition Components, U.S. Census Bureau.

Gómez, V., and A. Maravall, (1997), *TRAMO (Time Series Regression with ARIMA Noise, Missing Observations, and Outliers) and SEATS (Signal Extraction in ARIMA Time Series) Instructions for the User,* Banco de España, Madrid.

Gómez, V., and A. Maravall, (2001), Seasonal Adjustment and Signal Extraction in Economic Time Series, chapter 8 in D. Peña, G.C. Tiao, and R.S. Tsay (eds.), *A Course in Time Series Analysis,* John Wiley and Sons, New York.

Hillmer, S.C., and G.C. Tiao, (1982), An ARIMA-Model-Based Approach to Seasonal Adjustment, *Journal of the American Statistical Association,* 77, 63–70.

Kaiser, R., and A. Maravall, (2001), *Measuring Business Cycles in Economic Time Series,* Lecture Notes in Statistics 154, Springer-Verlag, New York.

Koopmans, S. J., Harvey, A. C., Doornik, J. A. and Shephard, N., (1995), *STAMP 5.0: Structural Time Series Analyser, Modeller and Predictor,* London: Chapman & Hall, 1995.

Ladiray, D., (2018), Calendar Effects, Chapter 5 in in Mazzi, G.L., D. Ladiray, and D.A. Rieser, (eds.), *Handbook on Seasonal Adjustment 2018 edition,* Eurostat: Publications Office of the European Union, Luxembourg.

Ladiray, D., and B. Quenneville, (2001), *Seasonal Adjustment with the X-11 Method,* Springer Lecture Notes in Statistics 158, Springer Verlag, Berlin.

Mazzi, G.L., D. Ladiray, and D.A. Rieser, (2018), *Handbook on Seasonal Adjustment, 2018 Edition,* Eurostat: Publications Office of the European Union, Luxembourg.

McElroy, T., and A. Roy, (2017), Detection of Seasonality in the Frequency Domain, Center for Statistical Research & Methodology Research and Methodology Directorate U.S. Census Bureau, Washington.

Pollock, D.S.G, (2016), Econometric Filters, *Computational Economics,* 48, 669–691.

Shiskin, J., A.H. Young, and J.C. Musgrave, (1967), *The X-11 Variant of the Census Method II Seasonal Adjustment,* Technical Paper No. 15, Bureau of the Census, U.S. Department of Commerce.

Taylor, C.J., (2017), *The CAPTAIN Toolbox for System Identification, Time Series Analysis, Forecasting and Control: Getting Started Guide.* Lancaster University.

Young, P.C., D.J. Pedregal, and W. Tych, (1999), Dynamic Harmonic Regression, *Journal of Forecasting,* 18, 369–394.