Defining a function file: Exponential smoothing filter example

The programming problem associated with the exponential smoothing filter is

$$\min_{\{y_t^g\}} \sum_{t=1}^{T}(y_t - y_t^g)^2 + \lambda \sum_{t=2}^{T}(y_t^g - y_{t-1}^g)^2$$

where y_t is the value of the series at date t and y_t^g is the growth component extracted by the filter. The first-order conditions for an interior solution to this problem are:

$$0 = -2(y_1 - y_1^g) - 2\lambda(y_2^g - y_1^g)$$

$$0 = -2(y_t - y_t^g) - 2\lambda[(y_{t+1}^g - y_t^g) - (y_t^g - y_{t-1}^g)]$$

for each $t \in \{2, \ldots, T-1\}$, and

$$0 = -2(y_T - y_T^g) + 2\lambda(y_T^g - y_{T-1}^g).$$

or

$$-\lambda y_{t+1}^g + (1+2\lambda)y_t^g - \lambda y_{t-1}^g = y_t$$

These may be viewed as a matrix system

$$Y = MY^g$$

with Y and Y^g being column vectors of size T .

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_t \\ \\ y_{T-1} \\ y_T \end{bmatrix} \qquad Y^g = \begin{bmatrix} y_1^g \\ y_2^g \\ \\ y_t^g \\ \\ y_{T-1}^g \\ y_T^g \end{bmatrix}$$

with

$$M=\begin{bmatrix} 1+\lambda & -\lambda & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\lambda & (1+2\lambda) & -\lambda & & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & \ddots & & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda & 1+2\lambda & -\lambda & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & \ddots & & \vdots \\ 0 & 0 & 0 & & 0 & 0 & 0 & & -\lambda & 1+2\lambda & -\lambda \\ 0 & 0 & 0 & & 0 & 0 & 0 & \cdots & 0 & -\lambda & 1+\lambda \end{bmatrix}.$$

The following function file constructs this matrix, taking in a vector Y and creating a vector Y^g , which is called "g" in the program.

esmf.m	
% applies the "Exponential Smoothing" filter to time series % using a matrix inversion technique. The data is assumed to be % organized into a matrix "y" with the rows of y % being the observations and the columns being the series. % The value of the smoothing parameter is the second argument. % The output is the filtered series g. function g=esmf(y,lam) % ESM filtering involves the solution of the difference equation: % y(t) = - lam*g(t+1) + (1+2*lam)*g(t) -lam*g(t-1) % with % y(1) = (1+lambda)*g(1) - lam*g(2) % y(T) = (1+lambda)*g(T) - lam*g(T-1) % If there are a smaller number of observations than series, then % there is most likely a mistake: convert the vector/matrix of inputs:	

<table>
<tr>
<td>

```
oy = size(y);
ny = max(oy); % length of time series
if (oy(1)<oy(2))
 y=y';
end



disp('Computing Exponential Smoothing Filtered
Time Series with Matrix Inversion')
disp('Growth Component is Returned as g')



% Strategy: Structure difference equation as a
matrix equation:
% M g = y
% and then invert M.



M = zeros(ny,ny);
d1=ones(ny-1,1);
d1=-d1*lam;
d2=ones(ny,1);
d2=d2*(1+2*lam);
d2(1)=1+lam;
d2(ny)=1+lam;
M = diag(d1,1)+diag(d2)+diag(d1,-1);
g = inv(M)*y;



% convert if necessary
if (oy(1)<oy(2))
 g=g';
end
```

</td>
<td>This program makes use of the built-in MATLAB function diag, to build the block diagonal matrix M. diag(x) puts the vector x on the main diagonal; diag(z,1) puts the vector z on the "subdiagonal"; diag(x,-1) puts the vector z on the "superdiagonal". z must be one element shorter than x, as in the code.</td>
</tr>
</table>