Derivation and Matlab/Octave implementation of Hodrick-Prescott filter

Grzegorz Klima*

February 20, 2007

1 Derivation

Hodrick-Prescott filter is a method of removing trend y_t^g from time series y_t. The trend is found as a solution to the following minimization problem:

$$\min_{(y_t^g)_{t=1,\ldots,T}} = \sum_{t=1}^{T}(y_t - y_t^g)^2 + \lambda \sum_{t=2}^{T-1}[(y_{t+1}^g - y_t^g) - (y_t^g - y_{t-1}^g)]^2, \qquad (1)$$

where λ is a penalty for trend's deviation from linear trend.

Rewrite our problem as:

$$\min_{(y_t^g)_{t=1,\ldots,T}} S((y_t^g)_{t=1,\ldots,T}) = \sum_{t=1}^{T}(y_t - y_t^g)^2 + \lambda \sum_{t=2}^{T-1}[y_{t+1}^g + y_{t-1}^g - 2y_t^g]^2$$

Assume $T > 4$. FOCs are (ignoring g_0 and g_{T+1}):

$$\frac{\partial S}{\partial y_1^g} = -2(y_1 - y_1^g) + 2\lambda(y_3^g + y_1^g - 2y_2^g) = 0,$$

$$\frac{\partial S}{\partial y_2^g} = -2(y_2 - y_2^g) + 2\lambda(y_4^g + y_2^g - 2y_3^g) - 4\lambda(y_3^g + y_1^g - 2y_2^g) = 0,$$

$$\frac{\partial S}{\partial y_t^g} = -2(y_t - y_t^g) + 2\lambda(y_t^g + y_{t-2}^g - 2y_{t-1}^g) - 4\lambda(y_{t+1}^g + y_{t-1}^g - 2y_t^g)$$

$$+2\lambda(y_{t+2}^g + y_t^g - 2y_{t+1}^g) = 0,\ 2 < t < T-1,$$

$$\frac{\partial S}{\partial y_{T-1}^g} = -2(y_{T-1} - y_{T-1}^g) + 2\lambda(y_{T-1}^g + y_{T-3}^g - 2y_{T-2}^g) - 4\lambda(y_T^g + y_{T-2}^g - 2y_{T-1}^g) = 0,$$

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

*Department of Economics, University of Minnesota. E-mail: klim0057@umn.edu.

Rearranging:

$$
\begin{gathered}
(\lambda+1)y_1^g - 2\lambda y_2^g + \lambda y_3^g = y_1, \\
-2\lambda y_1^g + (5\lambda+1)y_2^g - 4\lambda y_3 + \lambda y_4^g = y_2, \\
\lambda y_{t-2}^g - 4\lambda y_{t-1}^g + (6\lambda+1)y_t^g - 4\lambda y_{t+1}^g + \lambda y_{t+2}^g = y_t,\ 2 < t < T-1, \\
\lambda y_{T-3}^g - 4\lambda y_{T-2}^g + (5\lambda+1)2y_{T-1}^g - 2\lambda y_T^g = y_{T-1}, \\
\lambda y_{T-2}^g - 2\lambda y_{T-1}^g (\lambda+1)y_T^g = y_T.
\end{gathered}
$$

Since problem is quadratic FOCs reduce to a system of linear equations. The solution is our vector of trend values. In matrix form:[1]

$$
\begin{pmatrix}
\lambda+1 & -2\lambda & \lambda & & & & \\
-2\lambda & 5\lambda+1 & -4\lambda & \lambda & & & \\
\lambda & -4\lambda & 6\lambda+1 & -4\lambda & \lambda & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & \lambda & -4\lambda & 6\lambda+1 & -4\lambda & \lambda \\
 & & & \lambda & -4\lambda & 5\lambda+1 & -2\lambda \\
 & & & & \lambda & -2\lambda & \lambda+1
\end{pmatrix}
\begin{pmatrix} y_1^g \\ y_2^g \\ y_3^g \\ \vdots \\ y_{T-2}^g \\ y_{T-1}^g \\ y_T^g \end{pmatrix}
=
\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{T-2} \\ y_{T-1} \\ y_T \end{pmatrix}. \tag{2}
$$

Notice that the matrix on LHS is sparse/banded. If T were large it would be more efficient to take advantage of this fact and use special solver suited for this kind of problems.

2 Matlab/Octave implementation

Function `hpfilter` accepts data `y` and treats each column as a separate time series. There have to be at least 5 observations. The parameter `lambda` (second argument) can take on numeric values or be a character indicating annual, quarterly or monthly data. The choice of λ for those sampling

[1] In cases $T = 1, 2, 3, 4$ we would get:

$$
\begin{gathered}
y_1^g = y_1, \\
\begin{pmatrix} y_1^g \\ y_2^g \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \\
\begin{pmatrix} \lambda+1 & -2\lambda & \lambda \\ -2\lambda & 4\lambda+1 & -2\lambda \\ \lambda & -2\lambda & \lambda+1 \end{pmatrix}
\begin{pmatrix} y_1^g \\ y_2^g \\ y_3^g \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \\
\begin{pmatrix} \lambda+1 & -2\lambda & \lambda & \\ -2\lambda & 5\lambda+1 & -4\lambda & \lambda \\ \lambda & -4\lambda & 5\lambda+1 & -2\lambda \\ & \lambda & -2\lambda & \lambda+1 \end{pmatrix}
\begin{pmatrix} y_1^g \\ y_2^g \\ y_3^g \\ y_4^g \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix},
\end{gathered}
$$

but those situations are not interesting in any way.

frequencies follows [1]. `hpfilter` returns cyclical and the trend (growth) component of given series.

```
function [yc, yg] = hpfilter(y, lambda)
% function [yc, yg] = hpfilter(y, lambda)
%
% compute the HP filter for time series y
%
% inputs:
%  y - time series of at least 5 observations
%      each column treated as separate variable
%  lambda - a number or a character indicating
%         mothly ('m'), quaterly ('q') or annual ('a') data
%         choice of parameter is based on Ravn, Uhlig
%          "On Adjusting the HP-Filter for the Frequency of Observations",
%         Review of Economics and Statistics, vol. 84, 2002, pp. 371-76
% outputs:
%  yc - cyclical component
%  yc - trend (growth) component
%
% (c) Gregorz Klima 2007
%

if ((nargin < 2)||(nargin > 2))
        error("usage: hpfilter(y, lambda) - \
lambda must be a number or a character indicating \
mothly ('m'), quaterly ('q') or annual ('a') data");
end

if (ischar(lambda))
    if (lambda == 'm') l = 129600; end
    if (lambda == 'q') l = 1600; end
    if (lambda == 'a') l = 6.25; end
    if ((lambda ~= 'm') && (lambda ~= 'q') && (lambda ~= 'a'))
        error("usage: hpfilter(y, lambda) - \
lambda must be a number or a character indicating \
mothly ('m'), quaterly ('q') or annual ('a') data");
    end
else
    if ((size(lambda,1)~=1)||(size(lambda,2)~=1))
        error("usage: hpfilter(y, lambda) - \
lambda must be a number or a character indicating \
mothly ('m'), quaterly ('q') or annual ('a') data");
    else
```

```
        l = lambda;
    end
end

t = size(y, 1);

if (t < 5)
    error("hpfilter: time series must be at least 5 periods long");
end

A = diag(l*ones(1, t-2), -2) \
    - diag([2*l, 4*l*ones(1, t-3), 2*l], -1) \
    + diag([l+1, 5*l+1, (6*l+1)*ones(1, t-4), 5*l+1, l+1]) \
    - diag([2*l, 4*l*ones(1, t-3), 2*l], 1) \
    + diag(l*ones(1, t-2), 2);

yg = A \ y;
yc = y - yg;
```

References

[1] Ravn M. and Harald Uhlig. On adjusting the HP-filter for the frequency of observations. *Review of Economics and Statistics*, 84:371–76, 2002.