Reproducing the fully modified HP filter

Muhammad Yaseen

2026-11-05

Table of contents

1 Abstract **1**

2 The published values **1**

3 Weights (Figures 1 and 1a) **2**

4 Gains (Figures 2 and 2b) **4**

5 Loss (Figures 3 and 3b; Tables 3a and 3b) **6**

6 The filter on US data **8**

7 Agreement with the MATLAB code **11**

8 References **11**

1 Abstract

This vignette reproduces the published numbers of the fully modified HP filter of Hanif, Iqbal and Choudhary (2017): the weights of Figure 1 and 1a, the gains of Figures 2 and 2b, the losses of Figures 3 and 3b, and the end-point losses of Tables 3a and 3b. The losses of the HP filter and the annual column of the paper's own scheme are reproduced to the printed digits; the quarterly column of its scheme is not, and the vignette says why. It also checks that `fmhp_filter()` returns what the authors' MATLAB code returns.

2 The published values

Every number the package reproduces is a row of `inst/extdata/reference.csv`, with the call that recomputes it. The same file drives a test, so the check below runs at every `R CMD check`.

```
ref <-
  mhp_reference(
      source   = "Hanif"
    , evaluate = TRUE
    , level    = 1
    )
ref[, .(id, value, computed = round(x = computed, digits = 3), agrees)]
```

```
                        id value computed agrees
 1:    hanif2017_t3a_hp_T55 1.735    1.735   TRUE
 2:    hanif2017_t3a_hp_T54 1.733    1.733   TRUE
 3:    hanif2017_t3a_hp_T40 1.731    1.731   TRUE
 4:    hanif2017_t3b_hp_T72 1.757    1.757   TRUE
 5:   hanif2017_t3b_hp_T103 1.770    1.770   TRUE
 6:  hanif2017_t3a_ours_T55 0.626    0.626   TRUE
 7:  hanif2017_t3a_ours_T54 0.627    0.627   TRUE
 8:  hanif2017_t3a_ours_T50 0.623    0.623   TRUE
 9: hanif2017_fig1_hp_range 0.385    0.385   TRUE
10:   hanif2017_fig1a_range 0.182    0.182   TRUE
11:   hanif2017_t3b_ours_US 0.797    1.000  FALSE
```

```
ref[agrees == FALSE, note]
```

```
[1] "not reproduced: the quarterly column (0.770 to 0.827) is not given by the weights of th
```

3 Weights (Figures 1 and 1a)

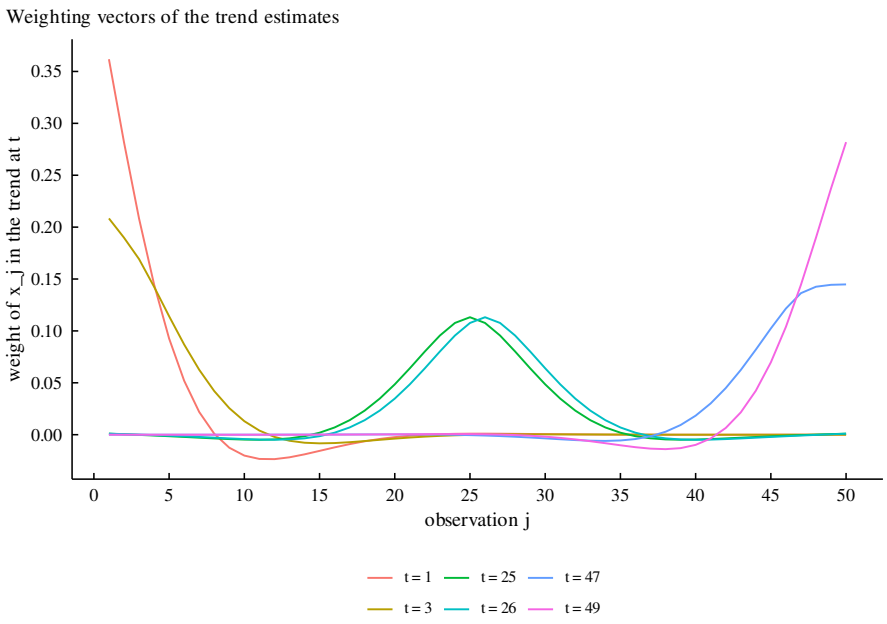
For the HP filter with $\lambda = 100$ and $T = 50$, the weights of the first estimate range over 0.385, from -0.024 to 0.362, the weight of the first observation in its own trend. The paper's Figure 1a shows the same weights with its quadratic scheme and a range of 0.182. That range is reached by the quadratic scheme with $k = 20$ and $\alpha = 10$, in the form of the MATLAB code (`form = "rows"`):

```
w_hp <-
  hp_weights(
      x      = 50
    , lambda = 100
    , form   = "rows"
    , rows   = NULL
    )
w_q <-
  hp_weights(
      x      = 50
    , lambda = hp_schedule(
          T      = 50
        , lambda = 100
        , alpha  = 10
```

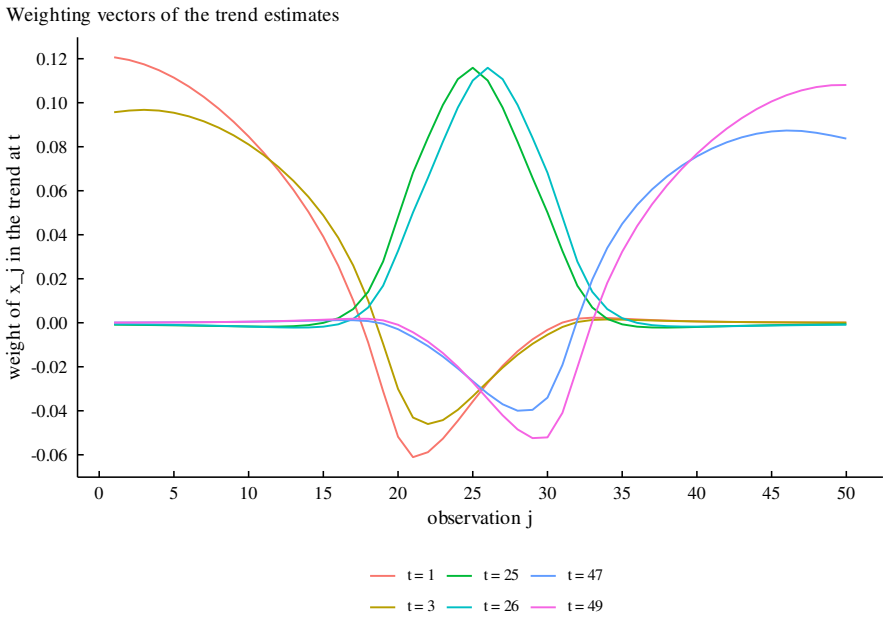
```
        , k       = 20
        , power   = 2
        , form    = "rows"
        )
    , form    = "rows"
    , rows    = NULL
    )
c(hp = diff(range(w_hp$weights[1, ])), quadratic =
 ↪  diff(range(w_q$weights[1, ])))
```

```
       hp quadratic
0.3853298 0.1817440
```

```
plot(x = w_hp)
```

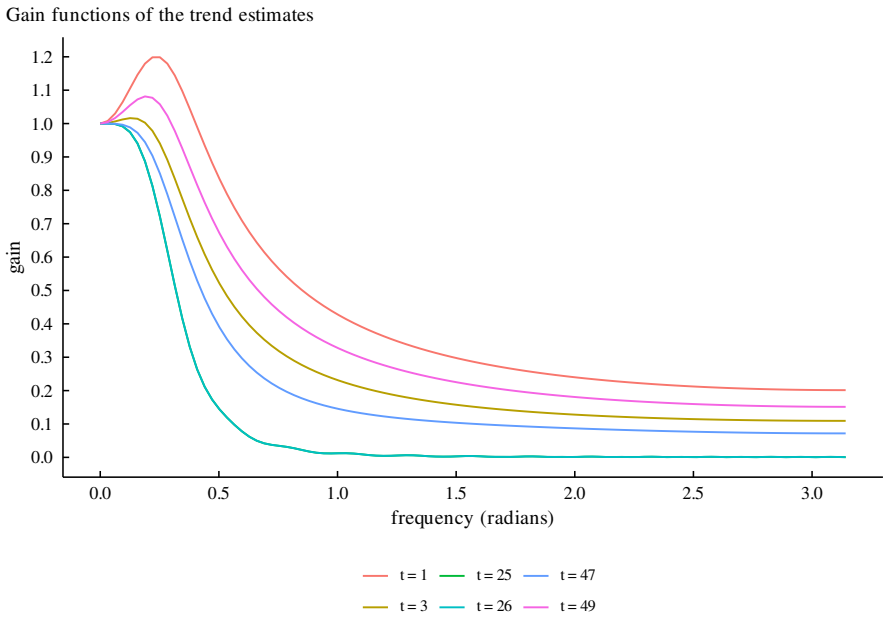



```
plot(x = w_q)
```

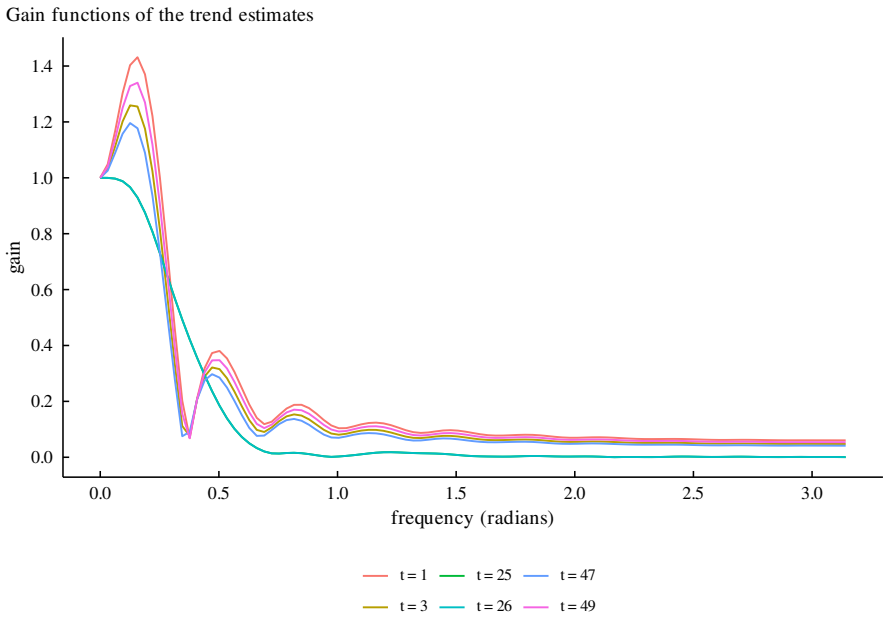


4 Gains (Figures 2 and 2b)

```
plot(
  x = hp_gain(
      x      = 50
    , lambda = 100
    , form   = "rows"
    , theta  = seq(from = 0, to = pi, length.out = 101)
    , rows   = NULL
    )
  )
```




```
plot(
  x = hp_gain(
      x      = 50
    , lambda = w_q$lambda_t
    , form   = "rows"
    , theta  = seq(from = 0, to = pi, length.out = 101)
    , rows   = NULL
    )
  )
```



With the end weights, the gains of the first and last estimates move towards the gain in the middle.

5 Loss (Figures 3 and 3b; Tables 3a and 3b)

```
l_hp <-
  hp_loss(
      x      = 55
    , lambda = 100
    , form   = "rows"
    , theta  = seq(from = 0, to = 3.1, by = 0.1)
    , weight = 0.1
    )
l_q <-
  hp_loss(
      x      = 55
    , lambda = hp_schedule(
          T      = 55
        , lambda = 100
        , alpha  = 10
        , k      = 20
        , power  = 2
        , form   = "rows"
        )
    , form   = "rows"
```

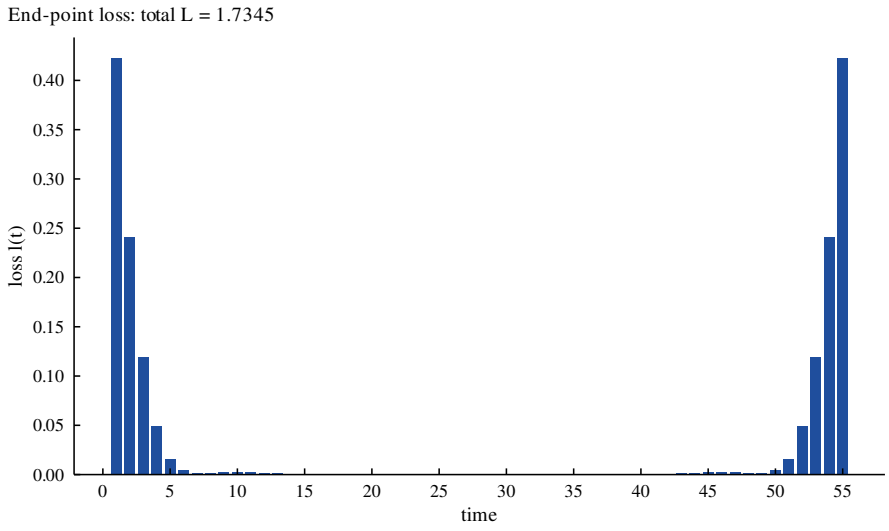
```
    , theta  = seq(from = 0, to = 3.1, by = 0.1)
    , weight = 0.1
    )
c(hp = l_hp$total, quadratic = l_q$total)
```

```
       hp quadratic
1.7345013 0.6256808
```

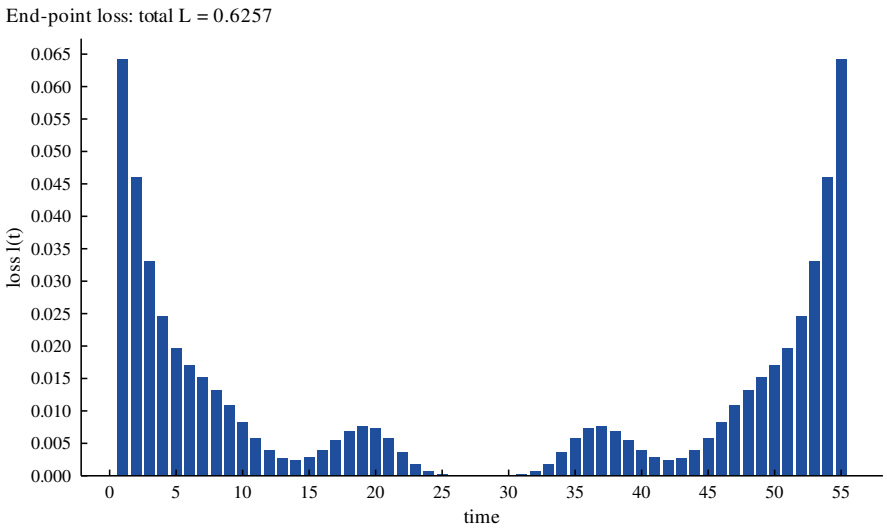
The loss depends only on T and the smoothing parameters, so the HP column of Tables 3a and 3b is a function of the length of each series: 1.731 to 1.735 for 40 to 55 annual values, 1.757 to 1.770 for 72 to 103 quarterly values. The package reproduces both once the centre of the loss is placed as MATLAB places it ($c = \lfloor T/2 + 0.5 \rfloor$; R's `round()` rounds halves to even and gives a different centre for odd T).

The column "HP (our scheme)" of Table 3a is reproduced for several series by the quadratic scheme of Figure 1a, $k = 20$ and $\alpha = 10$: 0.626 for $T = 55$, 0.627 for $T = 54$ and 0.623 for $T = 50$. Two values of that column (0.545 and 0.534) and the quarterly column of Table 3b (0.770 to 0.827) are not given by these weights: with $\lambda = 1600$ they give a loss of about 1.00. The paper does not report the weights or the lengths of the series behind those values, so they cannot be traced.

```
plot(x = l_hp)
```



```r
plot(x = l_q)
```



6 The filter on US data

`fmhp_filter()` runs the procedure of the paper: λ of the modified HP filter, the weight α of each scheme that minimises the loss, λ again with those weights, and the scheme with the smaller loss.

```r
fit <-
  fmhp_filter(
      x               = us_macro$gdp
    , max_lambda      = 3000
    , k_linear        = 15
    , k_quadratic     = NULL
    , alpha_linear    = seq(from = 0, to = 2000, by = 10)
    , alpha_quadratic = seq(from = 0, to = 2000, by = 5)
    , schemes         = c("linear", "quadratic")
    , form            = "rows"
    , lambda          = NULL
    , log_scale       = TRUE
    , as_dt           = FALSE
    )
fit
```

```
Fully modified HP Filter Result
===============================
Observations: 203
Fully modified lambda: 1448  (modified HP lambda: 683 )
```

```
End-point weights: quadratic, alpha = 15 on 22 end values; loss = 0.6163
GCV: 3.336e-04
Filtered on the log scale; cycle = series - trend in levels

Cycle Statistics:
  Mean: -2.603
  SD:   120.4
  AR1:  0.9522
  Range: 900

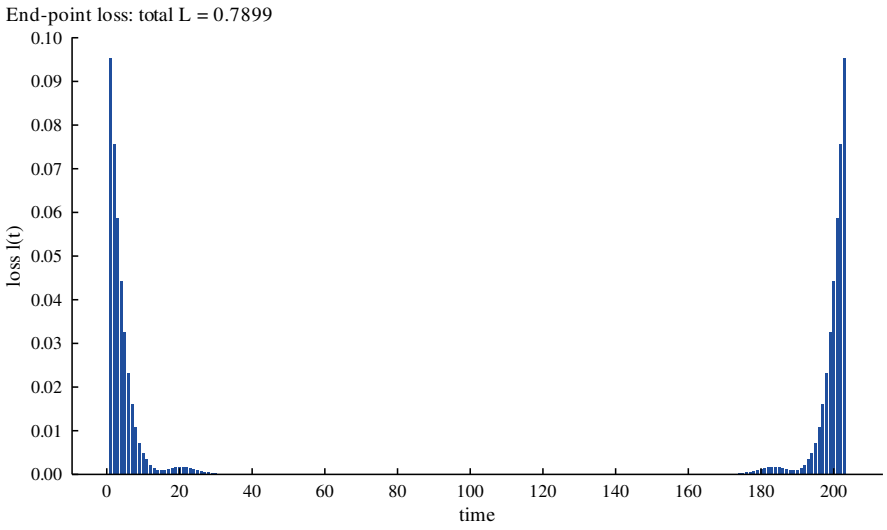
Sources: Hanif et al. (2017); Hanif (2017); Bloechl (2014); McDermott (1997);
Hodrick and Prescott (1997). Full references: mhp_cite(x).
```

The loss printed is the one by which the scheme was chosen, at λ of the modified HP filter, as in the MATLAB code (`minloss`). At the final λ the same weights give a larger loss:

```
hp_loss(
    x      = length(us_macro$gdp)
  , lambda = fit$lambda_t
  , form   = "rows"
  , theta  = seq(from = 0, to = 3.1, by = 0.1)
  , weight = 0.1
  )$total
```

```
[1] 0.7898641
```

```
ggplot2::autoplot(
    object = fit
  , which  = "loss"
  )
```



Over three US series:

```
fb <-
  fmhp_batch(
      X               = as.matrix(us_macro[, c("gdp", "consumption",
       ↪  "investment")])
    , max_lambda      = 3000
    , k_linear        = 15
    , k_quadratic     = NULL
    , alpha_linear    = seq(from = 0, to = 2000, by = 10)
    , alpha_quadratic = seq(from = 0, to = 2000, by = 5)
    , schemes         = c("linear", "quadratic")
    , form            = "rows"
    , lambda          = NULL
    , log_scale       = TRUE
    )
attr(x = fb, which = "lambdas")
```

```
        series lambda          gcv lambda_mhp    scheme alpha  k      loss
1:         gdp   1448 0.0003335623        683 quadratic    15 22 0.6163442
2: consumption   1261 0.0002248243        534 quadratic    15 22 0.5850792
3:  investment   2961 0.0073465762       1009 quadratic    20 22 0.6728679
```

The losses for the US in Table 3b of the paper lie between 0.539 and 0.566; ours are 0.59 to 0.67. They are not expected to agree: the paper's US series come from another source and cover another span, and the loss of the fully modified filter depends on the weights chosen, which depend on the data.

7 Agreement with the MATLAB code

The tests of the package carry a line-by-line port of the authors' MATLAB function (`tests/testthat/helper-matlab.R`, dense matrices and explicit loops) and compare it with `fmhp_filter()` on a simulated series: the smoothing parameters and the weights of both schemes agree exactly, their losses to 10^{-10}, and the trend and cycle to 10^{-8}. A port of the MATLAB code of the modified HP filter (Choudhary et al. 2014) is checked against `mhp_filter()` in the same way. The package reaches the same numbers by $O(T)$ banded solves instead of dense inverses.

8 References

Bloechl, A. (2014). Reducing the excess variability of the Hodrick-Prescott filter by flexible penalization. *Discussion Paper* 2014-1, University of Munich.

Hanif, M. N., Iqbal, J. and Choudhary, M. A. (2017). Fully modified HP filter. *SBP Working Paper Series*, No. 88, State Bank of Pakistan.

Choudhary, M. A., Hanif, M. N. and Iqbal, J. (2014). On smoothing macroeconomic time series using the modified HP filter. *Applied Economics*, 46(19), 2205–2214.

Hanif, M. N., Iqbal, J. and Choudhary, M. A. (2017). fmhpfilter: MATLAB code of the fully modified HP filter, version 1.0.0. MATLAB Central File Exchange 63198.