Software to perform temporal disaggregation of economic time series
A great deal of additional information on the European Union is available on the Internet. It can be accessed through the Europa server (http://europa.eu.int).

Luxembourg: Office for Official Publications of the European Communities, 2005

ISSN 1725-4825
ISBN 92-894-8626-0

© European Communities, 2005
Software to perform temporal disaggregation of economic time series

Ana Abad
Instituto Nacional de Estadística
c/ Rosario Pino, 14-16. Office 15.33
28020 - Madrid (SPAIN)
anaabad@ine.es

Enrique M. Quilis (corresponding author)
Instituto Nacional de Estadística
c/ Rosario Pino, 14-16. Office 15.34
28020 - Madrid (SPAIN)
emquilis@ine.es

Abstract

In this paper we present a program designed to perform temporal disaggregation of economic time series using a variety of techniques: univariate methods without indicators (Boot-Feibes-Lisman, Stram-Wei), univariate methods with indicators (Denton, Chow-Lin, Fernandez, Litterman, Santos-Cardoso, Guerrero) and multivariate methods with indicators and transversal constraints (Rossi, Denton, Di Fonzo). The program has two main components: a library of functions coded in Matlab and an Excel interface written in Visual Basic. This software is used to compile the Spanish Quarterly National Accounts (production mode) and to perform specific, detailed analysis that ensure the reliability and integrity of data, models and procedures (research mode).

1 The programs described in this paper have highly benefited from the comments and observations made by J. Bógalo, J.R. Cancelo, L. Navarro, S. Relloso, and by the participants of seminars on temporal disaggregation techniques held at the Instituto Nacional de Estadística, Instituto Gallego de Estadística and Universidad Autónoma de Madrid. The programs are freely available upon request.
1. INTRODUCTION

In this note we present an interface that enables the use of the temporal disaggregation library as described in Quilis (2004) under the Excel environment. Our main aim is that of combining the best features of both programs. On the one hand, the flexibility, power and easiness of modern worksheets has contributed to its de facto adoption as a reference for the storage and management of quantitative data-sets under the most varied circumstances, see Honoré and Poulsen (2002). On the other hand, programming languages oriented towards matrix, mathematical and symbolic manipulation enable economies of scale in the production and analysis of the underlying information, see LeSage (1999).

Basically, the interface consists of two main modules: a program which generates and manages a sequence of contextual menus and a linkage function that activates the temporal disaggregation library according to the user’s choices, as expressed by means of the corresponding forms. While the first module has been coded in Visual Basic, the second one has been written in the Matlab programming language.

This interface enables using environments in a simple and efficient way, permitting so quantitative analysts in general and, particularly, national accountants to easily integrate the temporal disaggregation techniques into their “tool-boxes”. In this sense, it also facilitates the transition from research mode to production mode, see Gatheral et al. (1999) for a detailed description of these issues.

2. DISAGGREGATION METHODS CONSIDERED


Although, the Stram-Wei (1986) and of the Guerrero (1990) methods have not been included due to their special information requisites, they remain directly accessible by means of the basic library through the functions \texttt{sw()} and \texttt{guerrero()} respectively.

The information-input requirements of the different univariate methods available, as well as the relevant constraints imposed on them have been detailed in the following table:

<table>
<thead>
<tr>
<th>Method</th>
<th>Inputs</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot-Feibes-Lisman</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Denton</td>
<td>X</td>
<td>p=1</td>
</tr>
<tr>
<td>Fernandez</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Chow-Lin</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Litterman</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Santos-Cardoso</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Multivariate methods enable the simultaneous estimation of a set of high-frequency data which have to satisfy a transversality condition. The information structure is summarized in the following table:

<table>
<thead>
<tr>
<th>Method</th>
<th>Inputs</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rossi</td>
<td>X</td>
<td>m=M</td>
</tr>
<tr>
<td>Denton</td>
<td>X</td>
<td>n=nz=s*N</td>
</tr>
<tr>
<td>Di Fonzo</td>
<td>X</td>
<td>m=M</td>
</tr>
</tbody>
</table>

\(^2\) The extension to the multivariate case of the Denton method is described in Di Fonzo (1994) and in Di Fonzo and Marini (2003).
3. STRUCTURE OF THE CONTEXTUAL MENUS

The initial menu allows selecting the type of temporal disaggregation technique: univariate or multivariate. It also grants access to specific information on the interface and on the Matlab library.

As a next step, in the univariate case, the user is asked to select the specific temporal disaggregation method to be employed and the corresponding required parameters. The menu itself will highlight the pertinent fields (i.e., "degree of differencing" if the Denton method is selected). The user has also to determine the frequency conversion procedure (i.e., annual to quarterly) as well as the nature of the transversal constraint (i.e., a flow’s distribution implies the summation of high-frequency data in order to obtain low-frequency ones).
The format of the output-file containing the relevant information, the determination of its level of detail and directory path are chosen in the next contextual menu:

Figure 3: Menu for the format and for the presentation of information. Univariate case.

The output-file can adopt three different formats: summarized, normal and detailed. In the first case, the output consists only in the high-frequency estimates. In the second, the estimation is accompanied by (if the method chosen permits doing so) its standard deviation, its intervals ±sigma and its residual series. Finally, if the user chooses the detailed mode, the output will consist in the normal output plus an ASCII-file containing the results of the estimated model and diverse diagnostic checks.

The scheme followed in the multivariate case is similar to the one described above. In the first place, the method, as well as its relevant parameters are selected:
Figure 4: Menu for the multivariate temporal disaggregation.

The selection of the relevant information and of the directory path of the output-file is carried out in the following menu:
Note that the series acting as the transversal constraint constitutes a critical (exogenous) input of the information set that has to be provided to the program. Optionally, the output can be accompanied by the standard errors of the high-frequency estimates.

Finally, in case the Di Fonzo method has been selected and various indicators are available for each low-frequency series, then a supplementary menu will be activated. In this menu the user must indicate the number of indicators available for each variable to be temporally disaggregated.
Appendix A: Univariate case: additional information

1. All the files of the system (Matlab.m functions, documents in pdf format, and this Excel file) must be allocated in the same directory, c:\td.

2. An additional directory, c:\td\output, must be created, to store an ASCII file that holds the information of the detailed output (model, parameters, correlations, etc.), when this option is selected.

3. It is essential to have access to Excel, Matlab, and its Excel Link toolbox must be operative.

4. Both Excel and Matlab must be active during execution, and Matlab must be activated through Excel (via Excel Link).

5. Restrictions on input data:
   - all the input series must be columnwise. The output series will also be columnwise arranged
   - an empty row must mark the end of input data
   - the series to be disaggregated must have a minimum of 3 observations
   - indicators must form a compact matrix (i.e., without empty columns among them)
   - all the indicators must have the same number of observations

6. Restrictions related to methods:
   - the Boot-Feibes-Lisman method does not use indicators; the Denton method requires only 1 indicator; the remaining methods require at least 1 indicator
   - the Denton method requires that n>=s*N; the remaining methods with indicator require that n>=s*N (n is the number of observations of the high-frequency indicator, s is the frequency conversion, and N is the number of observations of the low-frequency series)
   - the Chow-Lin, Litterman, and Santos-Cardoso methods allow to set the innovation parameter; the value of this parameter must lie between -1 and 1, both excluded

7. Output is written in the selected sheet, beginning in the selected initial cell. This cell marks the upper left corner of the data matrix formed by: estimate (brief output) or estimate, standard error, lower bound (estimate - s.e.), upper bound (estimate + s.e.) and residuals (normal and detailed output).

8. When the output normal or detailed is selected, the option Headlines writes the names of the output series in the line corresponding to the selected cell; data are written from the next line on.

9. Due to their intrinsic characteristics, the methods of Boot-Feibes-Lisman (BFL) and Denton do not generate standard errors, lower and upper limits, and BFL neither residuals. In order to preserve a common format for tabulation purposes, the interface fills this series with zeros, in normal and detailed output options.
Appendix B: Multivariante case: additional information

1. All the files of the system (Matlab.m functions, documents in pdf format, and this Excel file) must be allocated in the same directory, c:\td

2. It is essential to have access to Excel, Matlab, and its Excel Link toolbox must be operative

3. Both Excel and Matlab must be active during execution, and Matlab must be activated through Excel (via Excel Link)

4. Restrictions on input data:
   - all the input series must be columnwise. The output series will also be columnwise arranged
   - an empty row must mark the end of input data
   - all the series to be disaggregated must have the same number of observations, with a minimum of 3
   - indicators must form a compact matrix (i.e., without empty columns among them)
   - indicators must be arranged in the same order that the low frequency series
   - all the indicators must have the same number of observations
   - the transversal constraint must be just 1 series

5. Restrictions related to methods:
   - the methods of Rossi and Denton require that $\mathbf{m} = \mathbf{M}$; the Di Fonzo method requires that $\mathbf{m} \geq \mathbf{M}$ ($\mathbf{m}$ is the number of indicators, $\mathbf{M}$ is the number of aggregates); so, in the first case just 1 indicator is allowed for each aggregate, and in the last case each aggregate can have more than 1 indicator
   - the methods of Rossi and Denton require that $\mathbf{n} = \mathbf{nz} = \mathbf{s} \ast \mathbf{N}$; the Di Fonzo method requires that $\mathbf{n} \geq \mathbf{nz} \geq \mathbf{s} \ast \mathbf{N}$ ($\mathbf{n}$ is the number of observations of the indicators, $\mathbf{nz}$ is the number of observations of the transversal constraint, $\mathbf{s}$ is the frequency conversion, $\mathbf{N}$ is the number of observations of the aggregates); so, extrapolation is allowed only in the last case, with or without binding contemporaneous constraint

6. Output is written in the selected sheet, beginning in the selected initial cell. This cell marks the upper left corner of the data matrix formed by estimate series, in the same order of the input aggregates

7. If detailed information option in selected, the Di Fonzo method writes a matrix with the standard error of estimates, on the right side of the matrix with the estimates series, and in the same order. Due to their intrinsic characteristics, the methods of Rossi and Denton do not generate this kind of matrix.
Appendix C: A Matlab temporal disaggregation library

In this appendix we describe the syntax of all the functions contained in the temporal disaggregation library, including those not available through the Excel interface, e.g., `guerrero()`.

**C.1. BOOT-FEIBES-LISMAN**

PURPOSE: Temporal disaggregation using the Boot-Feibes-Lisman method

```
SYNTAX: res=bfl(Y,ta,d,s);
```

OUTPUT: res: a structure

- `res.meth` = 'Boot-Feibes-Lisman';
- `res.N` = Number of low frequency data
- `res.ta` = Type of disaggregation
- `res.s` = Frequency conversion
- `res.d` = Degree of differencing
- `res.y` = High frequency estimate
- `res.et` = Elapsed time

INPUT: Y: Nx1 ---> vector of low frequency data
- ta: type of disaggregation
  - ta=1 ---> sum (flow)
  - ta=2 ---> average (index)
  - ta=3 ---> last element (stock) ---> interpolation
  - ta=4 ---> first element (stock) ---> interpolation
- d: objective function to be minimized: volatility of ...
  - d=0 ---> levels
  - d=1 ---> first differences
  - d=2 ---> second differences
- s: number of high frequency data points for each low frequency data point
  - s= 4 ---> annual to quarterly
  - s=12 ---> annual to monthly
  - s= 3 ---> quarterly to monthly

LIBRARY: sw

SEE ALSO: tduni_print, tduni_plot

"Further methods of derivation of quarterly figures from annual data",

Application:

```matlab
Y=load('c:\x\td\data\Y.anu');
res=bfl(Y,1,1,12);
tduni_print(res,'td.sal');
tduni_plot(res);
edit td.sal
```
TEMPORAL DISAGGREGATION METHOD: Boot-Feibes-Lisman

Number of low-frequency observations: 22
Frequency conversion: 12
Number of high-frequency observations: 264
Degree of differencing: 1
Type of disaggregation: sum (flow).

High frequency series (columnwise):

4972.2800
4971.1389
........
........
7898.7692
7899.3631
7899.6600

Elapsed time: 0.3200
C.2. STRAM-WEI

PURPOSE: Temporal disaggregation using the Stram-Wei method.
-----------------------------------------------------------------------
SYNTAX: res = sw(Y,ta,d,s,v);
-----------------------------------------------------------------------
OUTPUT: res: a structure
       res.meth = 'Stram-Wei';
       res.N = Number of low frequency data
       res.ta = Type of disaggregation
       res.d = Degree of differencing
       res.s = Frequency conversion
       res.H = nxN temporal disaggregation matrix
       res.y = High frequency estimate
       res.et = Elapsed time
-----------------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
       ta: type of disaggregation
           ta=1 ---> sum (flow)
           ta=2 ---> average (index)
           ta=3 ---> last element (stock) ---> interpolation
           ta=4 ---> first element (stock) ---> interpolation
       d: number of unit roots
       s: number of high frequency data points for each low frequency data point
           s= 4 ---> annual to quarterly
           s=12 ---> annual to monthly
           s= 3 ---> quarterly to monthly
       v: (n-d)x(n-d) VCV matrix of high frequency stationary series
-----------------------------------------------------------------------
LIBRARY: aggreg, aggreg_v, dif, movingsum
-----------------------------------------------------------------------
SEE ALSO: bfl, tduni_print, tduni_plot
-----------------------------------------------------------------------
disaggregation of time series totals", Journal of Time Series Analysis,
vol. 7, n. 4, p. 293-302.

Application:

Y=load('c:\x\td\data\Y.anu');
N = length(Y); n = s*N;
% Defining the VCV matrix of stationary high-frequency time series
% Assumption of the example: IMA(d,2)
th1 = 0.9552; th2 = -0.0015; va = 0.87242 * ((223.5965)^2);
acf0 = va * (1+th1^2+th2^2); acf1 = -va * th1 * (1-th2); acf2 = -va * th2;
a0(1:n-d)=acf0; a1(1:n-d-1)=acf1; a2(1:n-d-2)=acf2;
v=diag(a0)+diag(a1,-1)+diag(a2,-2); v=v+tril(v)';
res = sw(Y,1,1,4,v);
tduni_print(res,'sw.sal');
tduni_plot(res);
edit sw.sal
TEMPORAL DISAGGREGATION METHOD: Stram-Wei

Number of low-frequency observations: 22
Frequency conversion: 4
Number of high-frequency observations: 88
Degree of differencing: 1
Type of disaggregation: sum (flow).

High frequency series (columnwise):

4792.4658
5015.8665
...
28880.7153
28822.8148

Elapsed time: 0.1100
C.3. DENTON

PURPOSE: Temporal disaggregation using the Denton method
-----------------------------------------------------------------------
SYNTAX: res=denton_uni(Y,x,ta,d,s);
-----------------------------------------------------------------------
OUTPUT: res: a structure
res.meth = 'Denton';
res.N = Number of low frequency data
res.ta = Type of disaggregation
res.s = Frequency conversion
res.d = Degree of differencing
res.y = High frequency estimate
res.x = High frequency indicator
res.U = Low frequency residuals
res.u = High frequency residuals
res.et = Elapsed time
-----------------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
x: nx1 ---> vector of low frequency data
ta: type of disaggregation
  ta=1 ---> sum (flow)
  ta=2 ---> average (index)
  ta=3 ---> last element (stock) ---> interpolation
  ta=4 ---> first element (stock) ---> interpolation
d: objective function to be minimized: volatility of ...
  d=0 ---> levels
  d=1 ---> first differences
  d=2 ---> second differences
s: number of high frequency data points for each low frequency data point
  s= 4 ---> annual to quarterly
  s=12 ---> annual to monthly
  s= 3 ---> quarterly to monthly
-----------------------------------------------------------------------
LIBRARY: aggreg, bfl
-----------------------------------------------------------------------
SEE ALSO: tduni_plot, tduni_print
-----------------------------------------------------------------------
REFERENCE: Denton, F.T. (1971) "Adjustment of monthly or quarterly
series to annual totals: an approach based on quadratic minimization",

Application:

Y=load('c:\x\td\data\Y.prn');
x=load('c:\x\td\data\x.ind');
res=denton_uni(Y,x,1,1,4);
tduni_plot(res,'td.sal');
tduni_print(res);
edit td.sal
ASCII file containing detailed output:

```
TEMPORAL DISAGGREGATION METHOD: Denton

Number of low-frequency observations : 22
Frequency conversion                  : 4
Number of high-frequency observations : 88
Degree of differencing                : 1
Type of disaggregation: sum (flow)

High frequency series (columnwise):
15374.9285
15169.7571
..........
24883.3098
20609.0705
24415.4509

Elapsed time: 0.0500
```
High frequency conformity

High frequency and low frequency residuals
C.4. CHOW-LIN

PURPOSE: Temporal disaggregation using the Chow-Lin method
-------------------------------------------------------------
SYNTAX: res=chowlin(Y,x,ta,s,type);
-------------------------------------------------------------
OUTPUT: res: a structure
  res.meth = 'Chow-Lin';
  res.ta = type of disaggregation
  res.type = method of estimation
  res.N = nobs. of low frequency data
  res.n = nobs. of high-frequency data
  res.pred = number of extrapolations
  res.s = frequency conversion between low and high freq.
  res.p = number of regressors (including intercept)
  res.Y = low frequency data
  res.x = high frequency indicators
  res.y = high frequency estimate
  res.y_dt = high frequency estimate: standard deviation
  res.y_lo = high frequency estimate: sd - sigma
  res.y_up = high frequency estimate: sd + sigma
  res.u = high frequency residuals
  res.U = low frequency residuals
  res.beta = estimated model parameters
  res.beta_sd = estimated model parameters: standard deviation
  res.beta_t = estimated model parameters: t ratios
  res.rho = innovational parameter
  res.aic = Information criterion: AIC
  res.bic = Information criterion: BIC
  res.val = Objective function used by the estimation method
  res.r = grid of innovational parameters used by the estimation method
-------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
  x: nxp ---> matrix of high frequency indicators (without intercept)
  ta: type of disaggregation
    ta=1 ---> sum (flow)
    ta=2 ---> average (index)
    ta=3 ---> last element (stock) ---> interpolation
    ta=4 ---> first element (stock) ---> interpolation
  s: number of high frequency data points for each low frequency data points
    s= 4 ---> annual to quarterly
    s=12 ---> annual to monthly
    s= 3 ---> quarterly to monthly
  type: estimation method:
    type=0 ---> weighted least squares
    type=1 ---> maximum likelihood
-------------------------------------------------------------
LIBRARY: aggreg
-------------------------------------------------------------
SEE ALSO: litterman, fernandez, td_plot, td_print
-------------------------------------------------------------
distribution and extrapolation of economic time series by related
Application:

```matlab
Y=load('c:\td\data\Y.prn');
x=load('c:\td\data\x.ind');
res=chowlin(Y,x,1,4,1);
td_print(res,'td.sal',1);  % op1=1: series are printed in ASCII file
td_plot(res);
edit td.sal
```

ASCII file containing detailed output:

```
TEMORAL DIAGGREGATION METHOD: Chow-Lin

Number of low-frequency observations :   22
Frequency conversion                 :    4
Number of high-frequency observations:   88
Number of extrapolations             :    0
Number of indicators (+ constant)    :    2

Type of disaggregation: sum (flow).

Estimation method: Maximum likelihood.

Beta parameters (columnwise):

* Estimate  * Std. deviation  * t-ratios
215.4518        111.7079          1.9287
0.9828          0.0069        142.0272

Innovational parameter:   0.7600

AIC:  10.0340
BIC:  10.1828

Low-frequency correlation
- levels  : 0.9998
- yoy rates  : 0.9617

High-frequency correlation
- levels  : 0.9998
- yoy rates  : 0.9812

High-frequency volatility of yoy rates
- estimate  : 8.4282
- indicator  : 9.0226
- ratio      : 0.9341
```
High frequency series (columnwise):
* Estimate
* Std. deviation
* 1 sigma lower limit
* 1 sigma upper limit
* Residuals

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. deviation</th>
<th>1 sigma lower limit</th>
<th>1 sigma upper limit</th>
<th>Residuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>5400.9896</td>
<td>114.8247</td>
<td>5286.1649</td>
<td>5515.8143</td>
<td>112.3095</td>
</tr>
<tr>
<td>5331.2409</td>
<td>83.7296</td>
<td>5227.5112</td>
<td>5394.9705</td>
<td>128.7034</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30079.6885</td>
<td>86.7557</td>
<td>29992.9328</td>
<td>30166.4443</td>
<td>-97.4913</td>
</tr>
<tr>
<td>25874.7702</td>
<td>86.2867</td>
<td>25788.4835</td>
<td>25961.0569</td>
<td>-43.9249</td>
</tr>
<tr>
<td>29614.4998</td>
<td>116.3242</td>
<td>29498.1756</td>
<td>29730.8240</td>
<td>-16.2417</td>
</tr>
</tbody>
</table>

Elapsed time: 1.8100
A variant to be applied with a fixed innovational parameter:

**PURPOSE:** Temporal disaggregation using the Chow-Lin method
rho parameter is fixed (supplied by the user)

**SYNTAX:** res=chowlin_fix(Y,x,ta,s,type,rho);
C.5. FERNANDEZ

PURPOSE: Temporal disaggregation using the Fernandez method

SYNTAX: res=fernandez(Y,x,ta,s);

OUTPUT: res: a structure
  res.meth  = 'Fernandez';
  res.ta    = type of disaggregation
  res.type  = method of estimation
  res.N     = nobs. of low frequency data
  res.n     = nobs. of high-frequency data
  res.pred  = number of extrapolations
  res.s     = frequency conversion between low and high freq.
  res.p     = number of regressors (including intercept)
  res.Y     = low frequency data
  res.x     = high frequency indicators
  res.y     = high frequency estimate
  res.y_dt  = high frequency estimate: standard deviation
  res.y_lo  = high frequency estimate: sd - sigma
  res.y_up  = high frequency estimate: sd + sigma
  res.u     = high frequency residuals
  res.U     = low frequency residuals
  res.beta  = estimated model parameters
  res.beta_sd = estimated model parameters: standard deviation
  res.beta_t = estimated model parameters: t ratios
  res.aic   = Information criterion: AIC
  res.bic   = Information criterion: BIC

INPUT: Y: Nx1  ---> vector of low frequency data
       x: nxp  ---> matrix of high frequency indicators (without intercept)
       ta: type of disaggregation
           ta=1  ---> sum (flow)
           ta=2  ---> average (index)
           ta=3  ---> last element (stock)  ---> interpolation
           ta=4  ---> first element (stock)  ---> interpolation
       s: number of high frequency data points for each low frequency data points
           s=4  ---> annual to quarterly
           s=12 ---> annual to monthly
           s=3  ---> quarterly to monthly

LIBRARY: aggreg

SEE ALSO: chowlin, litterman, td_plot, td_print

Application:

```matlab
Y = load('c:\x\td\data\Y.prn');
x = load('c:\x\td\data\x.tri');
res = fernandez(Y, x, 1, 4);
td_print(res, 'td.sal', 1); % op1=1: series are printed in ASCII file
td_plot(res);
edit td.sal
```

ASCII file containing detailed output:

```
******************************************************************************
TEMPORAL DISAGGREGATION METHOD: Fernandez
******************************************************************************

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of low-frequency observations</td>
<td>22</td>
</tr>
<tr>
<td>Frequency conversion</td>
<td>4</td>
</tr>
<tr>
<td>Number of high-frequency observations</td>
<td>90</td>
</tr>
<tr>
<td>Number of extrapolations</td>
<td>2</td>
</tr>
<tr>
<td>Number of indicators (+ constant)</td>
<td>2</td>
</tr>
</tbody>
</table>

Type of disaggregation: sum (flow).

Estimation method: Maximum likelihood.

Beta parameters (columnwise):

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. deviation</th>
<th>t-ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>564.9834</td>
<td>195.9404</td>
<td>2.8834</td>
</tr>
<tr>
<td>0.9360</td>
<td>0.0292</td>
<td>32.0284</td>
</tr>
</tbody>
</table>

Innovational parameter: 1.0000

AIC: 9.6079
BIC: 9.7567

Low-frequency correlation

- levels : 0.9998
- yoy rates : 0.9617

High-frequency correlation

- levels : 0.9997
- yoy rates : 0.9817

High-frequency volatility of yoy rates

- estimate : 8.3477
- indicator : 9.1506
- ratio : 0.9123
```
High frequency series (columnwise):

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. deviation</th>
<th>1 sigma lower limit</th>
<th>1 sigma upper limit</th>
<th>Residuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>5396.6742</td>
<td>91.6250</td>
<td>5305.0492</td>
<td>5488.2992</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>5297.9198</td>
<td>60.8871</td>
<td>5237.0327</td>
<td>5358.8069</td>
<td>2.3349</td>
<td></td>
</tr>
<tr>
<td>30021.1833</td>
<td>73.6977</td>
<td>29947.4856</td>
<td>30094.8810</td>
<td>920.9566</td>
<td></td>
</tr>
<tr>
<td>29586.1687</td>
<td>92.9937</td>
<td>29493.1750</td>
<td>29679.1625</td>
<td>1006.3644</td>
<td></td>
</tr>
<tr>
<td>28366.5459</td>
<td>140.8431</td>
<td>28225.7028</td>
<td>28507.3889</td>
<td>1006.3644</td>
<td></td>
</tr>
<tr>
<td>29461.6792</td>
<td>176.5235</td>
<td>29285.1557</td>
<td>29638.2027</td>
<td>1006.3644</td>
<td></td>
</tr>
</tbody>
</table>

Elapsed time: 0.0500

Graphs are the same than in the Chow-Lin case, except that the first one (objective function vs innovational parameter) is not generated.
C.6. LITTERMAN

PURPOSE: Temporal disaggregation using the Litterman method

------------------------------------------------------------
SYNTAX: res=litterman(Y,x,ta,s,type);
------------------------------------------------------------
OUTPUT: res: a structure
    res.meth    ='Litterman';
    res.ta      = type of disaggregation
    res.type    = method of estimation
    res.N       = nobs. of low frequency data
    res.n       = nobs. of high-frequency data
    res.pred    = number of extrapolations
    res.s       = frequency conversion between low and high freq.
    res.p       = number of regressors (including intercept)
    res.Y       = low frequency data
    res.x       = high frequency indicators
    res.y       = high frequency estimate
    res.y_dt    = high frequency estimate: standard deviation
    res.y_lo    = high frequency estimate: sd - sigma
    res.y_up    = high frequency estimate: sd + sigma
    res.u       = high frequency residuals
    res.U       = low frequency residuals
    res.beta    = estimated model parameters
    res.beta_sd = estimated model parameters: standard deviation
    res.beta_t  = estimated model parameters: t ratios
    res.rho     = innovational parameter
    res.aic     = Information criterion: AIC
    res.bic     = Information criterion: BIC
    res.val     = Objective function used by the estimation method
    res.r       = grid of innovational parameters used by the estimation method

------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
    x: nxp ---> matrix of high frequency indicators (without intercept)
    ta: type of disaggregation
        ta=1 ---> sum (flow)
        ta=2 ---> average (index)
        ta=3 ---> last element (stock) ---> interpolation
        ta=4 ---> first element (stock) ---> interpolation
    s: number of high frequency data points for each low frequency data points
        s= 4 ---> annual to quarterly
        s=12 ---> annual to monthly
        s= 3 ---> quarterly to monthly
    type: estimation method:
        type=0 ---> weighted least squares
        type=1 ---> maximum likelihood

------------------------------------------------------------
LIBRARY: aggreg

------------------------------------------------------------
SEE ALSO: chowlin, fernandez, td_plot, td_print

------------------------------------------------------------
REFERENCE: Litterman, R.B. (1983a) "A random walk, Markov model
for the distribution of time series", Journal of Business and
Application:

```matlab
Y=load('c:\td\data\Y.prn');
x=load('c:\td\data\x.tri');
res=litterman(Y,x,1,4,0);
td_print(res,'td.sal',0); % op1=0: series are not printed in ASCII file
td_plot(res);
edit td.sal
```

ASCII file containing detailed output:

```
******************************************************************************
TEMPORAL DISAGGREGATION METHOD: Litterman
******************************************************************************
Number of low-frequency observations :   22
Frequency conversion                 :    4
Number of high-frequency observations:   90
Number of extrapolations             :    2
Number of indicators (+ constant)    :    2
******************************************************************************
Type of disaggregation: sum (flow).
Estimation method: Weighted least squares.
******************************************************************************
Beta parameters (columnwise):
  * Estimate
  * Std. deviation
  * t-ratios

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. deviation</th>
<th>t-ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>1205.4851</td>
<td>233.5241</td>
<td>0.7910</td>
<td>0.0480</td>
</tr>
</tbody>
</table>

Innovational parameter:  0.9700
AIC:   7.9478
BIC:   8.0966
Low-frequency correlation
  - levels     : 0.9998
  - yoy rates  : 0.9617
High-frequency correlation
  - levels     : 0.9994
  - yoy rates  : 0.9735
High-frequency volatility of yoy rates
  - estimate   : 7.6249
  - indicator  : 9.1506
  - ratio      : 0.8333
Elapsed time:   2.5300
```
A variant to be applied with a fixed innovational parameter:

**PURPOSE:** Temporal disaggregation using the Litterman method
mu parameter is fixed (supplied by the user)

SYNTAX: res=litterman_fix(Y,x,ta,s,type,mu);

Graphical output contains the same information than in the Chow-Lin case.
C.7. SANTOS SILVA-CARDOSO

function res=ssc(Y,x,ta,s,type)
-------------------------------------------------------------
SYNTAX: res=ssc(Y,x,ta,s,type);
-------------------------------------------------------------
OUTPUT: res: a structure

  res.meth                ='Santos Silva-Cardoso';
  res.ta             = type of disaggregation
  res.type         = method of estimation
  res.N             = nobs. of low frequency data
  res.n              = nobs. of high-frequency data
  res.pred         = number of extrapolations
  res.s              = frequency conversion between low and high freq.
  res.p              = number of regressors (+ intercept)
  res.Y             = low frequency data
  res.x              = high frequency indicators
  res.y              = high frequency estimate
  res.y_dt          = high frequency estimate: standard deviation
  res.y_lo         = high frequency estimate: sd - sigma
  res.y_up        = high frequency estimate: sd + sigma
  res.u              = high frequency residuals
  res.U             = low frequency residuals
  res.gamma          = estimated model parameters (including y(0))
  res.gamma_sd    = estimated model parameters: standard deviation
  res.gamma_t             = estimated model parameters: t ratios
  res.rho         = dynamic parameter phi
  res.beta        = estimated model parameters (excluding y(0))
  res.beta_sd     = estimated model parameters: standard deviation
  res.beta_t      = estimated model parameters: t ratios
  res.aic         = Information criterion: AIC
  res.bic         = Information criterion: BIC
  res.val         = Objective function used by the estimation method
  res.r           = grid of dynamic parameters used by the estimation method
  res.et          = elapsed time

INPUT: Y: Nx1 ---> vector of low frequency data
x: nxp ---> matrix of high frequency indicators (without intercept)
ta: type of disaggregation
  ta=1 ---> sum (flow)
  ta=2 ---> average (index)
  ta=3 ---> last element (stock) ---> interpolation
  ta=4 ---> first element (stock) ---> interpolation
s: number of high frequency data points for each low frequency data points
  s= 4 ---> annual to quarterly
  s=12 ---> annual to monthly
  s= 3 ---> quarterly to monthly
  s= 4 ---> annual to quarterly
  s=12 ---> annual to monthly
  s= 3 ---> quarterly to monthly

LIBRARY: aggreg

SEE ALSO: chowlin, litterman, fernandez, td_plot, td_print

Application:

Y=load('c:\x\td\data\Y.prn');
x=load('c:\x\td\data\x.tri');
res=ssc(Y,x,1.4,1);
% Calling graph function
td_print(res,'td.sal',0);
edit td.sal;

ASCII file containing detailed output:

********************************************************************************
TEMPORAL DISAGGREGATION METHOD: Santos Silva-Cardoso
********************************************************************************

Number of low-frequency observations :   32
Frequency conversion                 :    4
Number of high-frequency observations:  128
Number of extrapolations             :    0
Number of indicators (+ constant)    :    2
Type of disaggregation: sum (flow).
Estimation method: Maximum likelihood.
Beta parameters (columnwise):
* Estimate    * Std. deviation    * t-ratios
1.0946        3.7817          0.2895
0.6718        0.0049          136.9983
Dynamic parameter: 0.2600
Long-run beta parameters (columnwise):
1.4792
0.9078
Truncation remainder: expected y(0):
* Estimate    * Std. deviation    * t-ratios
310.3328      90.5351          3.4278
AIC: 5.2524  
BIC: 5.3898  

Low-frequency correlation  
- levels : 0.9994  
- yoy rates : 0.8561  

High-frequency correlation  
- levels : 0.9993  
- yoy rates : 0.8881  

High-frequency volatility of yoy rates  
- estimate : 2.0592  
- indicator : 2.3430  
- ratio : 0.8789  

Graphical output contains the same information than in the Chow-Lin case and includes a plot of the implied impulse-response function:

![Impulse Response Function](image)

A variant to be applied with a fixed innovational parameter:

**PURPOSE:** Temporal disaggregation using the Santos Silva-Cardoso method  
Phi parameter is fixed (supplied by the user)  

**SYNTAX:** `res=ssc_fix(Y,x,ta,s,type,phi);`
C.8. GUERRERO

function res=guerrero(Y,x,ta,s,rexw,rexd);
PURPOSE: ARIMA-based temporal disaggregation: Guerrero method
---------------------------------------------------------------
SYNTAX: res=guerrero(Y,x,ta,s,rexw,rexd);
---------------------------------------------------------------
OUTPUT: res: a structure
  res.meth  = 'Guerrero';
  res.ta    = type of disaggregation
  res.N     = nobs. of low frequency data
  res.n     = nobs. of high-frequency data
  res.pred  = number of extrapolations
  res.s     = frequency conversion between low and high freq.
  res.p     = number of regressors (+ intercept)
  res.Y     = low frequency data
  res.x     = high frequency indicators
  res.w     = scaled indicator (preliminary hf estimate)
  res.y1    = first stage high frequency estimate
  res.y     = final high frequency estimate
  res.y_dt  = high frequency estimate: standard deviation
  res.y_lo  = high frequency estimate: sd - sigma
  res.y_up  = high frequency estimate: sd + sigma
  res.delta = high frequency discrepancy (y1-w)
  res.u     = high frequency residuals (y-w)
  res.U     = low frequency residuals (Cu)
  res.beta  = estimated parameters for scaling x
  res.k     = statistic to test compatibility
  res.et    = elapsed time
---------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
  x: nxp ---> matrix of high frequency indicators (without intercept)
  ta: type of disaggregation
    ta=1 ---> sum (flow)
    ta=2 ---> average (index)
    ta=3 ---> last element (stock) ---> interpolation
    ta=4 ---> first element (stock) ---> interpolation
  s: number of high frequency data points for each low frequency data points
    s= 4 ---> annual to quarterly
    s=12 ---> annual to monthly
    s= 3 ---> quarterly to monthly
  rexw, rexd ---> a structure containing the parameters of ARIMA model
                for indicator and discrepancy, respectively (see calT function)
---------------------------------------------------------------
LIBRARY: aggreg, calT, numpar, ols
---------------------------------------------------------------
SEE ALSO: chowlin, litterman, fernandez, td_print, td_plot
---------------------------------------------------------------
REFERENCE: Guerrero, V. (1990) "Temporal disaggregation of time
series: an ARIMA-based approach", International Statistical
Application:

Y=load('c:\xt\data\Y.prn');
x=load('c:\xt\data\x.tri');

% Inputs for td library
% Type of aggregation
ta=1;
% Frequency conversion
s=12;
% Model for w: (0,1,1)(1,0,1)
rexw.ar_reg = [1];
rexw.d = 1;
rexw.ma_reg = [1 -0.40];
rexw.ar_sea = [1 0 0 0 0 0 0 0 0 0 0 0 -0.85];
rexw.bd = 0;
rexw.ma_sea = [1 0 0 0 0 0 0 0 0 0 0 0 -0.79];
rexw.sigma = 4968.716^2;
% Model for the discrepancy: (1,2,0)(1,0,0)
% See: Martinez and Guerrero, 1995, Test, 4(2), 359-76.
rexd.ar_reg = [1 -0.43];
rexd.d = 2;
rexd.ma_reg = [1];
rexd.ar_sea = [1 0 0 0 0 0 0 0 0 0 0 0 0.62];
rexd.bd = 0;
rexd.ma_sea = [1];
rexd.sigma = 76.95^2;
% Calling the function: output is loaded in structure res
res=guerrero(Y,x,ta,s,rexw,rexd);
% Calling printing function
% Name of ASCII file for output
file_sal='guerrero.sal';
output=0; % Do not include series
% Calling graph function
td_plot(res);
### Temporal Disaggregation Method: Guerrero

- **Number of low-frequency observations**: 5
- **Frequency conversion**: 12
- **Number of high-frequency observations**: 60
- **Number of extrapolations**: 0
- **Number of indicators (+ constant)**: 2

Type of disaggregation: sum (flow).

Estimation method: BLUE.

#### Beta parameters (columnwise):

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. deviation</th>
<th>t-ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>219988.6766</td>
<td>974531.6756</td>
<td>4.4299</td>
<td></td>
</tr>
<tr>
<td>1723.8723</td>
<td>6174.6540</td>
<td>3.5819</td>
<td></td>
</tr>
</tbody>
</table>

AIC: 7.5245
BIC: 7.3683

Low-frequency correlation (Y,X)
- **levels**: 0.9003
- **yoy rates**: 0.9973

High-frequency correlation (y,x)
- **levels**: 0.9289
- **yoy rates**: 0.9835

High-frequency volatility of yoy rates
- **estimate**: 3.6623
- **indicator**: 6.2899
- **ratio**: 0.5823

High-frequency correlation (y,x*beta)
- **levels**: 0.9289
- **yoy rates**: 0.9832

Compatibility test:
- **k**: 0.9526
ARIMA model for scaled indicator:

\[
( 0 \ 1 \ 1 ) ( 1 \ 0 \ 1 )
\]

- Regular AR operator:
  1.0000

- Regular MA operator:
  1.0000 -0.4000

- Seasonal AR operator:
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 -0.8500

- Seasonal MA operator:
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 -0.7900

----------------------------------
ARIMA model for discrepancy:

\[
( 1 \ 2 \ 0 ) ( 1 \ 0 \ 0 )
\]

- Regular AR operator:
  1.0000 -0.4300

- Regular MA operator:
  1.0000

- Seasonal AR operator:
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.6200

- Seasonal MA operator:
  1.0000

----------------------------------

Elapsed time: 0.4400

----------------------------------

Graphical output contains the same information than in the Chow-Lin case.
C.9. ROSSI

function res = rossi(Y,x,z,ta,s,type);
PURPOSE: Multivariate temporal disaggregation with transversal constraint

----------------------------------------------------------------------------
SYNTAX: res = rossi(Y,x,z,ta,s,type);
----------------------------------------------------------------------------

OUTPUT: res: a structure
    res.meth = 'Multivariate Rossi';
    res.N   = Number of low frequency data
    res.n   = Number of high frequency data
    res.pred = Number of extrapolations (=0 in this case)
    res.ta  = Type of disaggregation
    res.s   = Frequency conversion
    res.y   = High frequency estimate
    res.et  = Elapsed time

----------------------------------------------------------------------------
INPUT: Y:  NxM  ---> M series of low frequency data with N observations
     x:  nxM  ---> M series of high frequency data with n observations
     z:  nx1  ---> high frequency transversal constraint
     ta: type of disaggregation
         ta=1  ---> sum (flow)
         ta=2  ---> average (index)
         ta=3  ---> last element (stock) ---> interpolation
         ta=4  ---> first element (stock) ---> interpolation
     s: number of high frequency data points for each low frequency data points
        s= 4  ---> annual to quarterly
        s=12 ---> annual to monthly
        s= 3  ---> quarterly to monthly
     type: univariate temporal disaggregation procedure used to compute
           preliminary estimates
        type = 1  ---> Fernandez
        type = 2  ---> Chow-Lin
        type = 3  ---> Litterman

LIBRARY: aggreg, vec, desvec, fernandez, chowlin, litterman

SEE ALSO: denton, difonzo, mtd_print, mtd_plot

time series when the aggregate is known", Review of Economics and Statistics,
vol. 64, n. 4, p. 695-696.
di Fonzo, T. (1994) "Temporal disaggregation of a system of
time series when the aggregate is known: optimal vs. adjustment methods",
Application:

```matlab
Y=load('YY.anu'); % Loading low frequency data
x=load('x.tri'); % Loading high frequency data
z=load('z.prn'); % Loading high frequency transversal restriction
res=rossi(Y,x,z,2,4,1);
mtd_print(res,'mtd.sal');
edit mtd.sal;
mtd_plot(res,z);
```

ASCII file containing detailed output:

```
TEMPORAL DISAGGREGATION METHOD: Multivariate Rossi

Number of low-frequency observations : 23
Frequency conversion                  : 4
Number of high-frequency observations : 92
Number of extrapolations      : 0

Type of disaggregation: average (index).

Preliminary univariate disaggregation: Fernandez

High frequency series (columnwise):
  * Point estimate

3424.2881  5311.2720
3436.0588  5280.4786
........  ........
........  ........
........  ........
2835.1833  8614.4139
2899.5740  8625.9809

Elapsed time:   1.2600
```
C.10. MULTIVARIATE DENTON

function res = denton(Y,x,z,ta,s,d);
PURPOSE: Multivariate temporal disaggregation with transversal constraint
-----------------------------------------------------------------------------------------------
SYNTAX: res = denton(Y,x,z,ta,s,d);
-----------------------------------------------------------------------------------------------
OUTPUT: res: a structure
res.meth = 'Multivariate Denton';
res.N = Number of low frequency data
res.n = Number of high frequency data
res.pred = Number of extrapolations (=0 in this case)
res.ta = Type of disaggregation
res.s = Frequency conversion
res.d = Degree of differencing
res.y = High frequency estimate
res.et = Elapsed time
-----------------------------------------------------------------------
INPUT: Y: NxM ---> M series of low frequency data with N observations
x: nxM ---> M series of high frequency data with n observations
z: nzx1 ---> high frequency transversal constraint
ta: type of disaggregation
    ta=1 ---> sum (flow)
    ta=2 ---> average (index)
    ta=3 ---> last element (stock) ---> interpolation
    ta=4 ---> first element (stock) ---> interpolation
s: number of high frequency data points for each low frequency data points
    s= 4 ---> annual to quarterly
    s=12 ---> annual to monthly
    s= 3 ---> quarterly to monthly
d: objective function to be minimized: volatility of ...
    d=0 ---> levels
    d=1 ---> first differences
    d=2 ---> second differences
-----------------------------------------------------------------------
LIBRARY: aggreg, aggreg_v, dif, vec, desvec
-----------------------------------------------------------------------
SEE ALSO: difonzo, mtd_print, mtd_plot
-----------------------------------------------------------------------
REFERENCE: di Fonzo, T. (1994) "Temporal disaggregation of a system of
time series when the aggregate is known: optimal vs. adjustment methods",
INSEE-Eurostat Workshop on Quarterly National Accounts, Paris, december
Application:

Y=load('YY.anu'); % Loading low frequency data
x=load('x.tri'); % Loading high frequency data
z=load('z.prn'); % Loading high frequency transversal restriction
res=denton(Y,x,z,2,4,1);
mtd_print(res,'mtd.sal');
edit mtd.sal;
mtd_plot(res,z);

ASCII file containing detailed output:

*******************************************************************************
TEMPORAL DISAGGREGATION METHOD: Multivariate Denton
*******************************************************************************

Number of low-frequency observations : 23
Frequency conversion                  : 4
Number of high-frequency observations : 92
Number of extrapolations              : 0

Degree of differencing                : 1
Type of disaggregation: average (index).

High frequency series (columnwise):
* Point estimate

3752.9096  4982.6505
3456.3681  5257.1693
.........  .........
.........  .........
.........  .........
2757.8458  8545.8074
2825.1411  8624.4561
2867.5816  8657.9733

Elapsed time: 0.2800
function res = difonzo(Y,x,z,ta,s,type,f);

PURPOSE: Multivariate temporal disaggregation with transversal constraint

SYNTAX: res = difonzo(Y,x,z,ta,s,type,f);

OUTPUT: res: a structure
- res.meth = 'Multivariate di Fonzo'
- res.N = Number of low frequency data
- res.n = Number of high frequency data
- res.pred = Number of extrapolations
- res.ta = Type of disaggregation
- res.s = Frequency conversion
- res.type = Model for high frequency innovations
- res.beta = Model parameters
- res.y = High frequency estimate
- res.d_y = High frequency estimate: std. deviation
- res.et = Elapsed time

INPUT: Y: NxM ---> M series of low frequency data with N observations
x: nxm ---> m series of high frequency data with n observations, m>=M see (*)
z: nzx1 ---> high frequency transversal constraint with nz obs.
ta: type of disaggregation
- ta=1 ---> sum (flow)
- ta=2 ---> average (index)
- ta=3 ---> last element (stock) ---> interpolation
- ta=4 ---> first element (stock) ---> interpolation
s: number of high frequency data points for each low frequency data points
- s= 4 ---> annual to quarterly
- s=12 ---> annual to monthly
- s= 3 ---> quarterly to monthly

type: model for the high frequency innovations
- type=0 ---> multivariate white noise
- type=1 ---> multivariate random walk

(*) Optional:
f: 1xM ---> Set the number of high frequency indicators linked to each low frequency variable. If f is explicitly included, the high frequency indicators should be placed in consecutive columns

NOTE: Extrapolation is automatically performed when n>sN.
If n=nz>sN restricted extrapolation is applied.
Finally, if n>nz>sN extrapolation is performed in constrained form in the first nz-sN observations and in free form in the last n-nz observations.

LIBRARY: aggreg, dif, vec, desvec

SEE ALSO: denton, mtd_print, mtd_plot

Application:

Y=load('YY.anu'); % Loading low frequency data
x=load('x.tri'); % Loading high frequency data
z=load('z.prn'); % Loading high frequency transversal restriction
res = difonzo(Y,x,z,2,4,1);
mtd_print(res,'mtd.sal');
edit mtd.sal;
mtd_plot(res,z);

ASCII file containing detailed output:

******************************************************************************
TEMPORAL DISAGGREGATION METHOD: Multivariate di Fonzo
******************************************************************************
Number of low-frequency observations : 23
Frequency conversion                  : 4
Number of high-frequency observations : 92
Number of extrapolations      : 0
******************************************************************************
Model for the innovations: random walk.
Type of disaggregation: average (index).
******************************************************************************
High frequency series (columnwise):
  * Point estimate
  3413.3839  5322.1762
  3447.4092  5269.1282
  ........    ........
  ........    ........
  ........    ........
  2758.4657  8545.1875
  2817.9882  8631.6090
  2856.1605  8669.3944

40
High frequency series (columnwise):

\[
\begin{align*}
\text{Std. desviation} \\
197.8732 & \quad 197.8732 \\
127.3900 & \quad 127.3900 \\
\ldots & \quad \ldots \\
137.9397 & \quad 137.9397 \\
128.1006 & \quad 128.1006 \\
194.9112 & \quad 194.9112 \\
\end{align*}
\]

Elapsed time: 0.3300
APPENDIX D: Additional notes

This release\(^3\) of the Matlab temporal disaggregation library includes some new features:

- stock first as a temporal disaggregation case (interpolation)
- new graphs for univariate Denton
- univariate Denton, proportional variant
- univariate temporal disaggregation by means of an ARIMA model-based procedure due to Guerrero (1990)
- multivariate temporal disaggregation by means of an two-step method due to Rossi (1982). The first step requires a preliminary univariate disaggregation that may be performed by Fernández, Chow-Lin or Litterman.

The library includes a set of function to perform temporal disaggregation (distribution, averaging and interpolation), according to the following structure:

**Adjustment or quadratic programming methods:**

- bfl (Boot-Feibes-Lisman)
- denton_uni, denton_uni_prop
- sw (Stram-Wei method)

    served by: tduni_print (ASCII output), tduni_plot (graphic output)

**Model-based (or BLUE) methods:**

- chowlin
- fernandez
- litterman
- ssc  (Santos Silva-Cardoso method: a dynamic version of Chow-Lin)

    served by: td_print (ASCII output), td_plot (graphic output)

- guererro

    served by: td_print_G (ASCII output), td_plot (graphic output)

**Multivariate methods that include a transversal restriction:**

- rossi
- denton
- difonzo

    served by: mtd_print (ASCII output), mtd_plot (graphic output)

Extrapolation is feasible using chowlin, fernandez, litterman, ssc and difonzo. Constrained extrapolation can be performed also by means of difonzo.

The presentation of the functions is self-contained: help text, script to run the function and output (ASCII file and plots).

This library is rather specific. Combining it with the Econometrics Toolbox of Professor James LeSage is a sensible decision. In fact, some procedures require to have access to it, although this dependence may be circumvented by appropriate code modification. For more information, consult his Internet site:

http://jpl.econ.utoledo.edu/faculty/lesage

---

APPENDIX E: RELATIONSHIPS AMONG FUNCTIONS IN THE LIBRARY

The “X → Y” notation means “X function calls Y function”.

- bfl → sw
- denton_uni → aggreg, bfl
- sw → aggreg, aggreg_v, dif, movingsum

- chowlin → aggreg
- fernandez → aggreg
- litterman → aggreg
- ssc → aggreg
- guerrero → aggreg, calT, numpar, ols(*)

- rossi → aggreg, vec, desvec, fernandez, chowlin, litterman
- denton → aggreg, aggreg_v, dif, vec, desvec
- difonzo → aggreg, dif, vec, desvec

- bal → vec, desvec
- td_print → tasa, aggreg
- td_print_G → tasa, aggreg, mprint(*)

- td_plot → tasa
- tduni_plot → temporal_agg

(*) From James Lesage's *Econometric Toolbox*
REFERENCES


