User Guide of GARCH-MIDAS and DCC-MIDAS MATLAB Programs

1. Introduction

The GARCH-MIDAS model decomposes the conditional variance into the short-run and long-run components. The former is a mean-reverting GARCH(1,1)-like process, while the latter is determined by a long history of the realized volatility or macroeconomic variables weighted by MIDAS polynomials.

The DCC-MIDAS model is a multivariate extension to the GARCH-MIDAS model with dynamic correlations. The DCC-MIDAS model decomposes the conditional covariance matrix into the variances and the correlation matrix, with a two-step model specification and estimation strategy. In the first step, conditional variances are estimated by the univariate GARCH-MIDAS models. In the second step, observations are deflated by the estimated mean and conditional variances, and the standardized residuals are thus constructed. The standardized residuals have a correlation matrix with GARCH-MIDAS-like dynamics. The long-run component is determined by the history of sample autocorrelations under MIDAS weights.

Following Engle, Ghysels and Sohn (2013), we specify a GARCH-MIDAS model by Eq (1) – (5):

$$r_{it} = \mu + \sqrt{\tau_t g_{it}}\varepsilon_{it}, \tag{1}$$

$$g_{it} = (1-\alpha-\beta) + \alpha\frac{(r_{i-1,t}-\mu)^2}{\tau_t} + \beta g_{i-1,t}, \tag{2}$$

$$\tau_t = m + \theta\sum_{k=1}^{K}\psi_k(\omega)V_{t-k}, \tag{3}$$

$$V_t = \sum_{i=1}^{N} r_{it}^2, \tag{4a}$$

$$V_t = \frac{1}{N}\sum_{i=1}^{N} x_{it}, \tag{4b}$$

$$\psi_k(\omega) \propto \left(1-\frac{k}{K}\right)^{\omega-1}, \tag{5a}$$

$$\psi_k(\omega) \propto \left(1-\frac{k}{K}\right)^{\omega_1-1}\left(\frac{k}{K}\right)^{\omega_2-1}, \tag{5b}$$

where r_{it} denotes the observation of day i in month t (or aggregation by weeks, quarters, years, etc.). The conditional variance is decomposed into the short-run component g_{it} and the long-run component τ_t. The former has a GARCH(1,1)-like recursion specified by Eq (2), while the latter is determined by the realized volatility or macroeconomic series. V_t in Eq (4a) is the realized volatility of the month, and V_t in Eq (4b) represents the monthly average of an exogenous variable (such as a monthly macroeconomic variable whose value x_{it} is fixed for $i = 1, \dots, N$). A long history of $V_{t-1}, V_{t-2}, \dots, V_{t-K}$ weighted by Beta polynomials (Eq (5a) or (5b)) captures the long-run volatility.

Colacito, Engle and Ghysels (2011) extends the model to the multivariate case. In the DCC-MIDAS model, the observations are m dimensional time series data, whose conditional covariance matrix is decomposed into m conditional variances and a $m \times m$ conditional correlation matrix, hence a two-step specification strategy. Each of the m conditional variances is assumed to follow a GARCH-MIDAS model.

The correlation matrix evolves over time. Consider a quasi-correlation matrix Q_t whose (i,j) element q_{ijt} has the dynamics

$$q_{ijt} = \rho_{ijt}(1-a-b) + a\varepsilon_{i,t-1}\varepsilon_{j,t-1} + bq_{ij,t-1}, \quad (6)$$

where $\varepsilon_{i,t-1}, \varepsilon_{j,t-1}$ are the standardized residuals of the previous period, so q_{ijt} has a GARCH(1,1)-like dynamics. The long-run component ρ_{ijt} is the (i,j) element of ρ_t, namely the MIDAS weighted-sum of the sample correlation matrices $c_{t-1}, c_{t-2}, \dots, c_{t-K}$.

$$\rho_t = \sum_{k=1}^{K} \psi_k(\omega) c_{t-k}, \quad (7)$$

where c_t is computed by the standard formula of the sample correlation matrix of length, say S.

The correlation matrix is a rescale of the quasi-correlation matrix so that the diagonals are unity:

$$R_t = [diag(Q_t)]^{-1/2} Q_t [diag(Q_t)]^{-1/2}. \quad (8)$$

2. Syntax

2.1 GarchMidas

`GarchMidas` is a MATLAB function for estimating a GARCH-MIDAS model. The syntax is

```
[...] = GarchMidas(y,name,value)
```

The required input argument is `y`, a $T \times 1$ vector of observations.

The optional name-value pairs include:

- `'X'`: T-by-1 macroeconomic data that determines the long-run conditional variance. If `X` is not specified, realized volatility will be used. `X` should be of the same length as `y`; repeat `X` values to match the date of `y` if necessary. Only one regressor is supported. The default is empty (realized volatility)
- `'Period'`: A scalar integer that specifies the aggregation periodicity (N). How many days in a week/month/quarter/year? How long is the secular component (τ_t) fixed? The default is 22 (as in a day-month aggregation)
- `'NumLags'`: A scalar integer that specifies the number of lags (K) in filtering the secular component by MIDAS weights. The default is 10 (say a history of 10 weeks/months/quarters/years)
- `'EstSample'`: A scalar integer that specifies a subsample `y(1:EstSample)` for parameter estimation. The remaining sample is used for conditional variance forecast and validation. The default is `length(y)`, no forecast.
- `'RollWindow'`: A logical value that indicates rolling window estimation on the long-run component. If true, the long-run component varies every period. If false, the long-run component will be fixed for a week/month/quarter/year. The default is false.
- `'LogTau'`: A logical value that indicates logarithmic long-run volatility component. The default is false.
- `'Beta2Para'`: A logical value that indicates two-parameter Beta MIDAS polynomial Eq (5b). The default is false (one-parameter Beta polynomial, Eq (5a)).

- `'Options'`: The `FMINCON` options for numerical optimization. For example,
  Display iterations: `optimoptions('fmincon','Display','Iter');`
  Change solver: `optimoptions('fmincon','Algorithm','active-set');`
  The default is the `FMINCON` default choice.
- `'Mu0'`: MLE starting value for the location-parameter (μ). The default is the sample average of observations.
- `'Alpha0'`: MLE starting value for α in the short-run GARCH(1,1) component. The default is 0.05.
- `'Beta0'`: MLE starting value for β in the short-run GARCH(1,1) component. The default is 0.9.
- `'Theta0'`: MLE starting value for the MIDAS coefficient $\sqrt{\theta}$ in the long-run component. If the name-value pair `'ThetaM'` is true, it is θ. The default is 0.1.
- `'W0'`: MLE starting value for the MIDAS parameter ω in the long-run component. The default is 5.
- `'M0'`: MLE starting value for the location-parameter $\sqrt{m}$ in the long-run component. If the name-value pair `'ThetaM'` is true, it is m. The default is 0.01.
- `'Gradient'`: A logical value that indicates analytic gradients in MLE. The default is false.
- `'AdjustLag'`: A logical value that indicates MIDAS lag adjustments for initial observations due to missing presample values. The default is false.
- `'ThetaM'`: A logical value that indicates not taking squares for the parameter theta and m in the long-run volatility component. The default is false (they are squared).
- `'Params'`: Parameter values for $(\mu, \alpha, \beta, \theta, \omega, m)$. In that case, the program will skip MLE, and just infer the conditional variances based on the specified parameter values. The default is empty (need parameter estimation).
- `'ZeroLogL'`: A vector of indices between 1 and T, which select a subset of dates and forcefully reset the likelihood values of those dates to zero. For example, use `ZeroLogL` to ignore initial likelihood values. The default is empty (no reset).

The output arguments include:

- `estParams`: Estimated parameters for $(\mu, \alpha, \beta, \theta, \omega, m)$.
- `EstParamCov`: Estimated parameter covariance matrix.
- `Variance`: T-by-1 conditional variances.
- `LongRunVar`: T-by-1 long-run component of conditional variances.
- `ShortRunVar`: T-by-1 short-run component of conditional variances.
- `logL`: T-by-1 log likelihood. Initial observations may be assigned a flag of zero.

2.2 DccMidas

`DccMidas` is a MATLAB function for estimating a DCC-MIDAS model. The syntax is

```
[...] = DccMidas(Data,name,value)
```

The required input argument is `Data`, a $T \times m$ matrix of observations.

The optional name-value pairs include:

- `'Period'`: A scalar integer that specifies the aggregation periodicity (N). How many days in a week/month/quarter/year? How long is the secular component (τ_t) fixed? The default is 22 (as in a day-month aggregation)
- `'NumLagsVar'`: A scalar integer that specifies the number of lags (K) in filtering the secular component by MIDAS weights. This is for the first step GARCH-MIDAS model. The default is 10 (say a history of 10 weeks/months/quarters/years)
- `'NumLagsCorr'`: A scalar integer that specifies the number of lags(K) in filtering the secular component by MIDAS weights. This is for the second step estimation of correlation matrix. The default is 10 (say a history of 10 weeks/months/quarters/years)
- `'EstSample'`: A scalar integer that specifies a subsample `y(1:EstSample)` for parameter estimation. The remaining sample is used for conditional variance forecast and validation. The default is `length(y)`, no forecast.
- `'RollWindow'`: A logical value that indicates rolling window estimation on the long-run component. If true, the long-run component varies every period. If false, the long-run component will be fixed for a week/month/quarter/year. The default is false.
- `'LogTau'`: A logical value that indicates logarithmic long-run volatility component. This is for the first step GARCH-MIDAS model. The default is false.
- `'Beta2Para'`: A logical value that indicates two-parameter Beta MIDAS polynomial, Eq (5b). The default is false (one-parameter Beta polynomial, Eq (5b)).
- `'Options'`: The `FMINCON` options for numerical optimization. For example,
  Display iterations: `optimoptions('fmincon','Display','Iter');`
  Change solver: `optimoptions('fmincon','Algorithm','active-set');`
  The default is the `FMINCON` default choice.
- `'Mu0'`: MLE starting value for the location-parameter (μ). The default is the sample average of observations.
- `'Alpha0'`: MLE starting value for α in the short-run GARCH(1,1) component. The default is 0.05.
- `'Beta0'`: MLE starting value for β in the short-run GARCH(1,1) component. The default is 0.9.
- `'Theta0'`: MLE starting value for the MIDAS coefficient $\sqrt{\theta}$ in the long-run component. If the name-value pair `'ThetaM'` is true, it is θ. The default is 0.1.
- `'W0'`: MLE starting value for the MIDAS parameter ω in the long-run component. The default is 5.
- `'M0'`: MLE starting value for the location-parameter $\sqrt{m}$ in the long-run component. If the name-value pair `'ThetaM'` is true, it is m.The default is 0.01.
- `'CorrA0'`: MLE starting value for a in the GARCH(1,1) component. It is either a scalar (if all variables share it) or a column vector (if each variable has its own parameter). This is for the second step correlation matrix estimation. The default is 0.05 (or a vector expansion).
- `'CorrB0'`: MLE starting value for b in the GARCH(1,1) component. It is either a scalar (if all variables share it) or a column vector (if each variable has its own parameter). This is for the second step correlation matrix estimation. The default is 0.05 (or a vector expansion).
- `'CorrW0'`: MLE starting value for the MIDAS parameter w in the long-run component. It is a scalar. Vector is not supported. The default is 0.05.
- `'MorePara'`: A logical value that indicates multivariate series have different a, b. However, the program only supports a single ω. This is for the second step correlation matrix estimation. The default is false (parameters a, b, ω are shared by all variables)

- `'Gradient'`: A logical value that indicates analytic gradients in MLE. The default is false.
- `'AdjustLag'`: A logical value that indicates MIDAS lag adjustments for initial observations due to missing presample values. The default is false.
- `'ThetaM'`: A logical value that indicates not taking squares for the parameter theta and m in the long-run volatility component. The default is false (they are squared).
- `'ZeroLogL'`: A vector of indices between 1 and T, which select a subset of dates and forcefully reset the likelihood values of those dates to zero. For example, use `ZeroLogL` to ignore initial likelihood values. The default is empty (no reset).

The output arguments include:

- `estParamsStep1`: 6-by-n estimated parameters for $(\mu, \alpha, \beta, \theta, \omega, m)$, obtained from the univariate GARCH-MIDAS models.
- `EstParamCovStep1`: 6-by-6-by-n estimated parameter covariance matrix, obtained from the univariate GARCH-MIDAS models
- `estParamsStep2`: 3-by-1 or (2n+1)-by-1 estimated parameters, obtained from the second-step autocorrelation matrix estimation.
- `EstParamCovStep2`: 3-by-3 or (2n+1)-by-(2n+1) estimated parameter covariance matrix, obtained from the second-step autocorrelation matrix estimation.
- `Variance`: T-by-n conditional variances.
- `LongRunVar`: T-by-n long-run component of conditional variances.
- `CorrMatrix`: n-by-n-by-T conditional correlation matrices.
- `LongRunCorrMatrix`: n-by-n-by-T long-run component of the correlation matrices.
- `logL`: T-by-1 log likelihood. Initial observations may be assigned a flag of zero.

3. Examples

3.1 A GARCH-MIDAS Example

We downloaded the NASDAQ Composite Index daily return data (1971 – 2015) from the FRED Economic Data (NASDAQCOM). Though our data are not the same as those used in Engle, Ghysels and Sohn (2013), we try if we could obtain similar volatility results after 1970s.

To run the program, we could simply type `GarchMidas(y)` and accept all the default settings. However, there are some name-value pairs we may want to fine tune. `'Period'` specifies aggregation periodicity. If we put 22, it is roughly a day-month aggregation. `'NumLags'` specifies the number of MIDAS lags. Here we put 24, meaning a history of 24 months' realized volatility will be averaged by the MIDAS weights to determine the long-run conditional variance.

```
FILE    NAVIGATE    EDIT    BREAKPOINTS    RUN
e1.m  ×  +
% NASDAQ Composite Index, daily percentage change 1971 - 2015
% Data Source: FRED database
% https://research.stlouisfed.org/fred2/series/NASDAQCOM
y = xlsread('NASDAQCOM.xls','B22:B11669') ./ 100;

% Estimate the GARCH-MIDAS model, and extract the volatilities
period = 22;
numLags = 24;
[estParams,EstParamCov,Variance,LongRunVar] = GarchMidas(y,'Period',period,'NumLags',numLags)
```

```
Command Window
  Method: Maximum likelihood
  Sample size: 11648
  Adjusted sample size: 11120
  Logarithmic  likelihood:       36393.4
  Akaike   info criterion:      -72774.9
  Bayesian info criterion:      -72730.7
                   Coeff          StdErr        tStat      Prob
                ----------     ----------     -------     -----

      mu        0.00080314     7.3864e-05     10.873       0
      alpha        0.12607      0.0043966     28.674       0
      beta         0.81026      0.0083922     96.549       0
      theta         0.1849      0.0050338     36.733       0
      w             5.8269        0.68289     8.5328       0
      m          0.0050642     0.00025503     19.858       0

fx >>
```

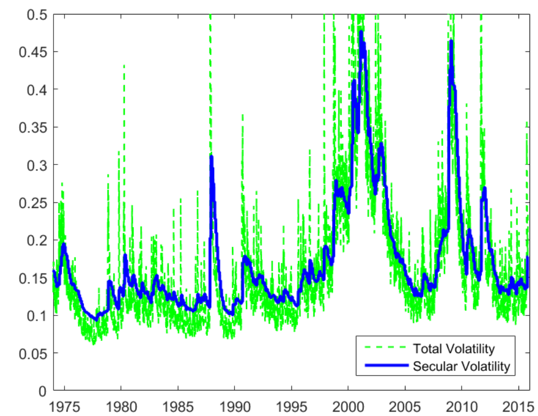
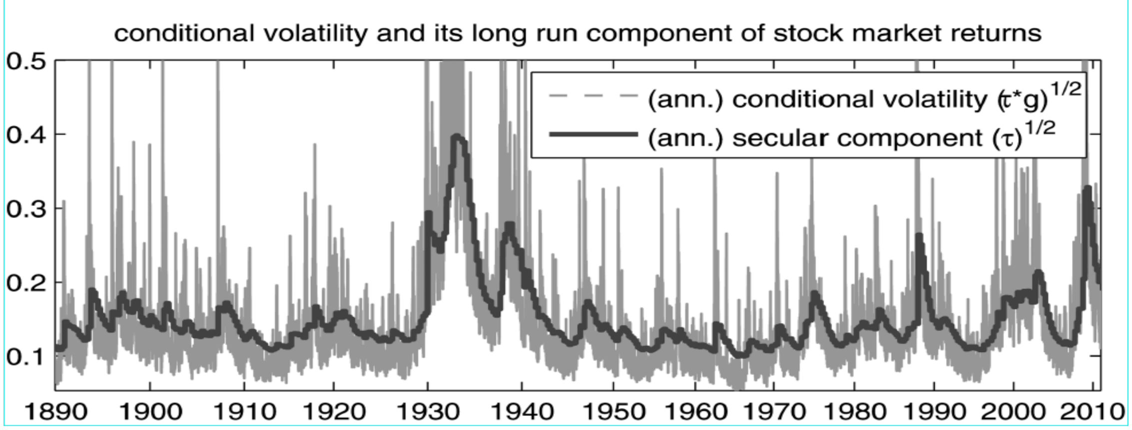


FIGURE 2.—GARCH-MIDAS WITH FIXED SPAN RV, 1890–2010



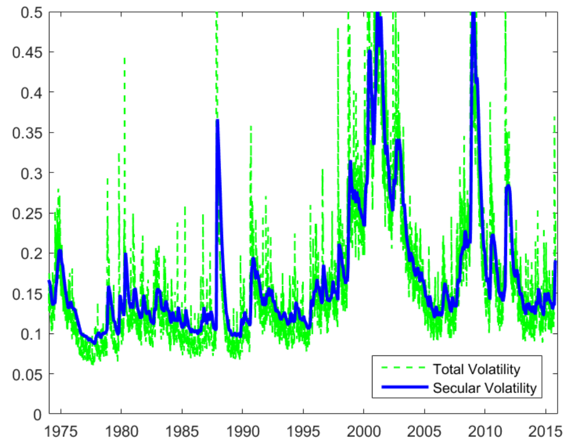
Source: Figure 2 of Engle, Ghysels and Sohn (2013)

Our estimated conditional volatility and its secular component in 1975 – 2010 have similar patterns as those reported in Figure 2 of Engle, Ghysels and Sohn (2013). The long-run component exhibits spikes in years around 1975, 1989, 2002, 2008, etc. The total volatility jumps upwards during those recession periods. It confirms the empirical regularity of the countercyclical stock market volatility.

The rolling window specification of the long-run conditional variance uses a different weight sequence for the realized volatility. To check whether it will produce similar results or not, we may run the program with the name-value pair `'RollWindow'`. The codes run a little slower due to more MIDAS weighed terms, but the results appear close to those under the fixed window specification.

```
% Estimate the rolling window version of the GARCH-MIDAS model
[estParams,EstParamCov,Variance,LongRunVar]...
    = GarchMidas(y,'Period',period,'NumLags',numLags,'RollWindow',1);
```

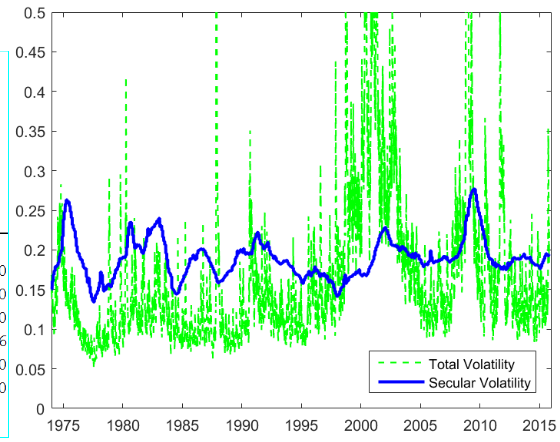
```
Method: Maximum likelihood
Sample size: 11648
Adjusted sample size: 11120
Logarithmic  likelihood:      36399.9
Akaike   info criterion:     -72787.7
Bayesian info criterion:     -72743.5
               Coeff          StdErr        tStat     Prob
            __________     __________     _______     ____

    mu      0.00080289     7.4187e-05      10.823      0
    alpha      0.13281      0.0048218      27.543      0
    beta        0.7822       0.010471      74.701      0
    theta      0.19053      0.0043393      43.908      0
    w           8.7578        0.96896      9.0383      0
    m        0.0045943     0.00022618      20.312      0

fx >>
```



The realized volatility could be a noisy proxy for the macro-volatility. We may replace the realized volatility by some direct measure of economic activities. We downloaded the Industrial Production Index growth rate data (1971-2015) from the FRED database (INDPRO). The program requires the exogenous variable formatted as a vector with the same length as the observation series `y`. So we just repeat the monthly values throughout the days. Then we can run the program with the name-value pair `'X'`.

```
% Industrial Production Index growth rate, 1971-2015
% Data Source: FRED database
% https://research.stlouisfed.org/fred2/series/INDPRO
xMonth = xlsread('INDPRO.xls','B42:B576') ./ 100;

% Repeat the monthly value throughout the days in that month
[~,yDate] = xlsread('NASDAQCOM.xls','A22:A11669');
[~,yDateMonth] = datevec(yDate);
xDay = NaN(nobs,1);
count = 1;
for t = 1:nobs
    if t > 1 && yDateMonth(t) ~= yDateMonth(t-1)
        count = count + 1;
        if count > length(xMonth)
            break
        end
    end
    xDay(t) = xMonth(count);
end

% Estimate the rolling window version of the GARCH-MIDAS model
[estParams,EstParamCov,Variance,LongRunVar] = GarchMidas(y,'Period',period,'NumLags',32,'X',xDay);
```

```
Method: Maximum likelihood
Sample size: 11648
Adjusted sample size: 10944
Logarithmic  likelihood:        35774
Akaike   info criterion:      -71536.1
Bayesian info criterion:      -71491.9
                 Coeff          StdErr        tStat          Prob
             ___________    ____________    ________    ____________

    mu        0.00077109      7.2524e-05      10.632               0
    alpha         0.1022       0.0030644      33.352               0
    beta         0.88682       0.0033269      266.56               0
    theta      -0.014074       0.0029087     -4.8384      1.3089e-06
    w             2.4424         0.41713      5.8552               0
    m         0.00016871      2.2356e-05      7.5465               0

fx >>
```



Lastly, we do some forecast exercise. We may run a subsample estimation and leave some observations for the one-step forecast validation by setting the name-value pairs `'EstSample'`. For example, we use 8000 observations for parameter estimation and the remaining observations for forecast validation. The software reports on the screen the root mean squared errors (RMSE) of the one-step forecast on the conditional variance.

```
% In-sample forecast validation
GarchMidas(y,'Period',period,'NumLags',numLags,'estSample',8000);
```

```
Method: Maximum likelihood
Sample size: 8000
Adjusted sample size: 7472
Logarithmic  likelihood:      25254.1
Akaike   info criterion:      -50496.2
Bayesian info criterion:      -50454.2
                Coeff          StdErr        tStat      Prob
             ___________    ____________    ________    ____

    mu        0.00088309      8.2387e-05      10.719     0
    alpha        0.17105       0.0068514      24.966     0
    beta         0.71813        0.013063      54.975     0
    theta        0.19193       0.0050618      37.918     0
    w             8.4068         0.99459      8.4526     0
    m          0.0043922      0.00023089      19.023     0

RMSE of one-step variance forecast (period 1 to 8000): 4.558e-04.
RMSE of one-step variance forecast (period 8001 to 11648): 5.160e-04.
```

We may want to perform out-of-sample volatility forecast. Eq (2) specifies the conditional variance recursion: $g_{it} = (1-\alpha-\beta) + \alpha\frac{(r_{i-1,t}-\mu)^2}{\tau_t} + \beta g_{i-1,t}$. Note that g_{it} is a deterministic function of $r_{i-1,t}$ and historical observations. For out-of-sample forecast, $\left(r_{i-1,t}-\mu\right)^2$ is not available. We may replace such unavailable observations by the forecasted variance, similar to the way we iteratively forecast an autoregressive process. We may call `GarchMidas` recursively to forecast future variances.

```
% Out-of-sample forecast
estParams = GarchMidas(y,'Period',period,'NumLags',numLags);
nForecast = 5;
yBig = [y;0];
for t = 1:nForecast
    [~,~,Variance,LongRunVar] = GarchMidas(yBig,'Period',period,'NumLags',numLags,'Params',estParams)
    yPseudo = estParams(1) + sqrt(Variance(end));
    yBig = [yBig(1:end-1);yPseudo;0];
end
VarianceForecast = Variance(nobs+1:nobs+nForecast);
LongRunVarForecast = LongRunVar(nobs+1:nobs+nForecast);
```

3.2 A DCC-MIDAS example

In this example, we try to use the `DccMidas` program to reproduce the results in Colacito, Engle and Ghysels (2011). The tri-variate DCC-MIDAS model consists of Energy and Hi-Tech portfolios and a 10 year bond. Users are responsible for obtaining their original data. Alternatively, the program will load a different dataset containing the NASDAQ daily returns, JPY/USD exchange rates percentage change and 10-Year treasury rates percentage change, downloaded from FRED Economic Data (NASDAQCOM, DEXJPUS, DGS10, respectively).

To use the software, users may simply type `DccMidas(Data)`. Similar to `GarchMidas`, setting some of the name-value pairs may be helpful. `'Period'` specifies aggregation periodicity. If we put 22, it is roughly a day-month aggregation. `'NumLagsVar'` specifies the number of MIDAS lags for the univariate GARCH-MIDAS for the first-step variance estimation. Here we put 36, meaning a history of 36 months' realized volatility will be averaged by the MIDAS weights to determine the long-run conditional variance. `'NumLagsCorr'` specifies the number of MIDAS lags for the second-step correlation matrix estimation. We put lagged values of 144 months in this application, but users may reduce the number of lags if the sample size is smaller.

To reproduce the results of the paper, we will overload some of the default name-value pairs of `DccMidas`, because their results were estimated by different codes. `'Options'` is the `FMINCON` options for numerical optimization. We use the legacy `'active-set'`, though the default choice is `'interior-point'`. Also, by setting `'ZeroLogL'` to 1:3600, we forcefully suppress the contribution of the initial 3600 observations to the likelihood function, though the default initialization scheme does not have a burn-in of that amount. Such adjustment is for compatibility with others' implementation on the DCC-MIDAS model. Also, we reset the MLE starting values `'mu0'` to 0.001. Numerical optimization will not work well unless starting values are carefully chosen.

```
% Estimate the DCC-MIDAS model
options = optimoptions('fmincon','Algorithm','active-set');
[estParamsStep1,~,estParamsStep2,~,Variance,LongRunVar,CorrMatrix,LongRunCorrMatrix]...
    = DccMidas(Data,'Period',20,'NumLagsVar',36,'NumLagsCorr',144,'options',options,'ZeroLogL',1:3600,'mu0',0.001);
CorrMatrix = reshape(CorrMatrix,9,nobs)';
LongRunCorrMatrix = reshape(LongRunCorrMatrix,9,nobs)';
```

The program first estimates three univariate GARCH-MIDAS models for the conditional variances, and then constructs the standardized residuals and estimates the correlation matrix.

```
Method: Maximum likelihood
Sample size: 8827
Adjusted sample size: 8107
Logarithmic  likelihood:     -12291.1
Akaike   info criterion:      24594.3
Bayesian info criterion:      24636.8
```

	Coeff	StdErr	tStat	Prob
mu	0.06978	0.011112	6.2798	0
alpha	0.088307	0.0031895	27.687	0
beta	0.79673	0.016579	48.055	0
theta	0.19945	0.0049751	40.089	0
w	13.356	1.7771	7.5158	0
m	0.54037	0.0328	16.475	0

```
Method: Maximum likelihood
Sample size: 8827
Adjusted sample size: 8107
Logarithmic  likelihood:       -13535
Akaike   info criterion:      27082.1
Bayesian info criterion:      27124.6
```

	Coeff	StdErr	tStat	Prob
mu	0.062998	0.012912	4.8791	1.0659e-06
alpha	0.086857	0.0030831	28.172	0
beta	0.83684	0.014522	57.624	0
theta	0.18674	0.0064655	28.882	0
w	10.169	1.8541	5.4846	0
m	0.72539	0.045025	16.111	0

```
Method: Maximum likelihood
Sample size: 8827
Adjusted sample size: 8107
Logarithmic  likelihood:     -7797.56
Akaike   info criterion:      15607.1
Bayesian info criterion:      15649.6
```

	Coeff	StdErr	tStat	Prob
mu	0.022186	0.0059381	3.7362	0.00018686
alpha	0.059198	0.0033665	17.584	0
beta	0.91586	0.0066424	137.88	0
theta	0.20377	0.0065901	30.92	0
w	2.9872	0.73878	4.0434	5.2681e-05
m	0.29003	0.028859	10.05	0

```
Method: Two-Step Maximum likelihood
Sample size: 8827
Adjusted sample size: 5227
Logarithmic  likelihood:     -21598.1
Akaike   info criterion:      43202.1
Bayesian info criterion:      43223.4
```

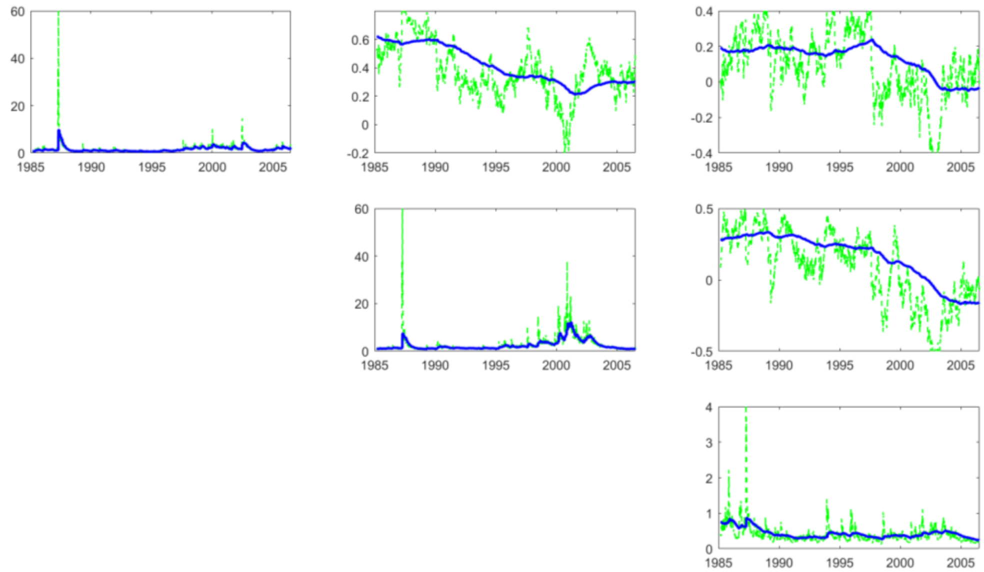
	Coeff	StdErr	tStat	Prob
a	0.017591	0.0013486	13.044	0
b	0.97741	0.001994	490.18	0
w	1.8246	0.70672	2.5818	0.0098302

The program nearly reproduces the results; both the estimated parameters and the volatility estimation are close to Table 1 and Figure 1 of Colacito, Engle and Ghysels (2011).

	mu	alpha	beta	theta	w	m
Energy	0.06978	0.088307	0.79673	0.19945	13.356	0.54037
Hi-Tech	0.062998	0.086857	0.83684	0.18674	10.169	0.72539
Bond	0.022186	0.059198	0.91586	0.20377	2.9872	0.29003

	a	b	w
DCC-MIDAS	0.017591	0.97741	1.8246

```
>>
```



If the users cannot obtain the original data used by the paper, the program will load an alternative dataset consisting of stock returns, exchange rate returns and bond yields percentage changes. Using the same codes, the estimation results are the following:

```
Method: Maximum likelihood
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:      -10909
Akaike   info criterion:     21829.9
Bayesian info criterion:     21872.7
```

	Coeff	StdErr	tStat	Prob
mu	0.086574	0.0079921	10.832	0
alpha	0.15278	0.0059163	25.823	0
beta	0.7429	0.01208	61.5	0
theta	0.2036	0.0047777	42.614	0
w	7.8053	0.82288	9.4853	0
m	0.39976	0.023648	16.905	0

```
Method: Maximum likelihood
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:    -7296.97
Akaike   info criterion:     14605.9
Bayesian info criterion:     14648.7
```

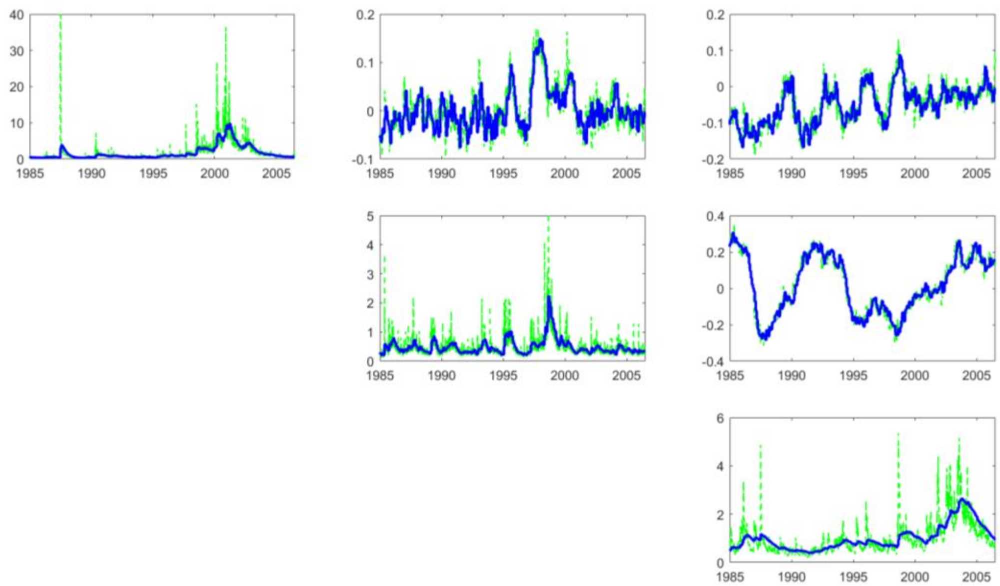
	Coeff	StdErr	tStat	Prob
mu	-0.001992	0.0056466	-0.35278	0.72426
alpha	0.14489	0.0032909	44.028	0
beta	0.68896	0.010587	65.076	0
theta	0.23074	0.0025254	91.369	0
w	12.267	0.62979	19.479	0
m	0.16631	0.0094764	17.55	0

```
Method: Maximum likelihood
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:    -10295.3
Akaike   info criterion:     20602.7
Bayesian info criterion:     20645.4
```

	Coeff	StdErr	tStat	Prob
mu	0.0021314	0.0072295	0.29482	0.76813
alpha	0.055403	0.0029636	18.695	0
beta	0.91936	0.0056068	163.97	0
theta	0.22148	0.0054202	40.862	0
w	2.7536	0.53081	5.1875	0
m	0.24627	0.036903	6.6735	0

```
Method: Two-Step Maximum likelihood
Sample size: 9236
Adjusted sample size: 5636
Logarithmic  likelihood:    -24124.3
Akaike   info criterion:     48254.7
Bayesian info criterion:     48276.1
```

	Coeff	StdErr	tStat	Prob
a	0.0077311	0.002346	3.2954	0.00098303
b	0.97311	0.0231	42.126	0
w	14.176	2.6517	5.346	0



4. Usage Notes and Tips

- ✓ The program requires MATLAB Optimization Toolbox, Statistics and Machine learning Toolbox. It works best for MATLAB 2015b, but it may work slowly under previous versions.
- ✓ Users may want to run the codes using different MLE starting values and compare the likelihood function values to determine the maximum likelihood estimator. It is good practice to reset the name-value pairs for the starting values.
- ✓ In case of poor MLE results, possibly with a warning messages `'Covariance matrix of estimators cannot be computed precisely...'`, try to refine the starting values and rescale the data. Also, setting the name-value pair `'Gradient'` may help.
- ✓ In case of error messages such as `'FMINCON failed...'`, the most likely cause is the conditional variance at some date is not positive at the starting parameter values. Try to change the starting values, and make changes to model specification if necessary.

Contact Author: Hang Qian, matlabist@gmail.com