Additional information by GAS program GasDuration

This document gives additional information users should know before implementing the program GasDurationUniv.ox. The class of duration models covered by the program is given by

$$
\begin{aligned}
y_t &= \lambda(f_t)\varepsilon_t, \\
f_t &= \omega + \sum_{i=1}^{p} A_i s_{t-i} + \sum_{j=1}^{q} B_j f_{t-j}, \qquad t = 1, \dots, n,
\end{aligned}
\tag{1}
$$

with $\{\varepsilon_t\}$ a sequence of independent and identically distributed non-negative random variables with $\mathbb{E}(\varepsilon_t) = 1$, $\lambda(f_t)$ a link function and s_t the scaled score. The parameter vector θ is given by

$$
\theta = (\omega, A_1, \dots, A_p, B_1, \dots, B_q, \alpha, \kappa), \tag{2}
$$

where α is a shape parameter which is estimated if ε_t is Weibull, Gamma, Generalised Gamma or Log-Normal distributed. The shape parameter κ is estimated in combination with α and is estimated if the selected distribution is Generalised Gamma. The user of the program is referred to Creal, Koopman, and Lucas (2013) for more explanation on GAS models.

User input

The user input is located between line 270 and 288 of the program. The following code is copied from the program.

```
270 mdata = loadmat("3m9912-adur.xls"); s_mY = mdata[][0]';
271 dscaling = 1; // Scaling data can improve stability, 1 for no scaling
272 s_mY.*= dscaling;
273 // Distribution: EXP, WEI, GAM, GEN_GAM, LOG_NORM
274 s_iDistribution = EXP;
275 // Link function: LAMBDA (f_t=lambda_t), LOG_LAMBDA (f_t=log(lambda_t))
276 s_iLinkFunction = LAMBDA;
277 // Scaling score: INV_FISHER, INV_SQRT_FISHER
278 s_iScalingChoice = INV_FISHER;
279 // Order of GAS model p, q
280 s_ip = 1; s_iq = 1;
281 // Standard erros: HESS, SAND
282 s_iStdErr = HESS;
283 // Starting values (note the dimensions of s_ip and s_iq)
284 domega = 0.05;
285 vA = <0.12>'; // Extend for higher orders of p, separate with comma's.
286 vB = <0.92>'; // Extend for higher orders of q, separate with comma's.
                     B_1 + ... + B_q < 1
287 dalpha = 0.18; // Shape parameter
288 dkappa = 1; // Shape parameter
```

The user input starts by loading the data which needs to be analysed (code line 270). The dataset `3m9912-adur.xls`, available from the same source this document comes, is loaded as example. In some cases the optimising process becomes more stable if the data is scaled by a factor (code line 271).

1. Choice of disturbance density, (code line 274): available choices are `EXP`, `WEI`, `GAM`, `GEN_GAM` and `LOG_NORM`.

2. Choice of link function $\lambda(f_t)$ (code line 276): available choices are `LAMBDA` ($f_t = \lambda_t$) and `LOG_LAMBDA` ($f_t = \log \lambda_t$).

3. Choice of scaling of the score, (code line 278): available choices are `INV_FISHER` and `INV_SQRT_FISHER`.

4. Order of the GAS model, (code line 280): available choices are any integer > 0 with a maximum dependent on what the data can identify.

5. Choice of standard error type, (code line 282): available choices are `HESS` (empirical Hessian) and `SAND` (sandwich estimator).

6. Starting values for the maximising algorithm, (code line 284 to 288): if the link function is specified as `LAMBDA`, the parameter ω is restricted to be $\omega \geq 0$ which is guaranteed by a log transformation of the parameter in the model. No actions for this are required by the user. The user needs to extend the vector of starting values for `vA` (code line 285) and `vB` (code line 286) to the number equal to `s_ip` and `s_iq` (code line 280), respectively. The sum of the elements in `vB` cannot exceed 1. Note that obtaining a global maximum is not always guaranteed and trying different starting values could be useful in some situations.

Computational details

1. Standard errors of the MLE are calculated by inverting the numerically computed Hessian matrix and applying the delta method to the transformed parameter(s).

2. The unconditional mean of f_t is used as initial condition given by $f_0 = \omega(1 - B)^{-1}$.

3. The first `max(s_ip, s_iq)` observations do not contribute directly to the likelihood function as described in, for example, Tsay (2005) p107.

Model output

1. The program output is the estimated duration λ_t in the top panel, the score ∇_t in the mid panel and the scaled score $s_t = S_t \nabla_t$ in the bottom panel, all for $t = 1, \ldots, n$ where S_t is the scaling matrix which depends on the choice of the user.

2. Activate the function `CompareDuration` (code line 306) to estimates the parameter vector for two selected densities (code lines 304, 305) and plot the estimated durations in one graph.

Example

We illustrate the working of the model with an example. The user input starts at line 270 and ends at line 288. This program comes with a data set of adjusted intraday trading durations of 3M stock in December 1999. The data does not need to be scaled as it won't give any problems with estimating the parameter vector. We start the analysis of the data by selecting the following options

```
s_iDistribution = EXP;
s_iLinkFunction = LAMBDA;
s_iScalingChoice = INV_FISHER;
s_ip = 1; s_iq = 1;
s_iStdErr = HESS;
```

and starting values

```
domega = 0.05;
vA = <0.12>';
vB = <0.92>';
```

Since we selected the option `s_iDistribution = EXP`, starting values for `dalpha` and `dkappa` do not need to be specified (any number will do). After running the program the output should say

```
Strong convergence using numerical derivatives
Log likelihood value = -3042.963812
Parameters with standard errors:
omega     0.17395 (0.10786)
A1        0.12134 (0.04076)
B1        0.9237  (0.04751)
```

The program should converge in around 31 iterations which takes between one and two seconds on a modern desktop pc. The output window should be like the one showed in Figure 1. Next, we change the distribution to the Generalised Gamma distribution. For this we change the input block to

```
s_iDistribution = GEN_GAM;
s_iLinkFunction = LAMBDA;
s_iScalingChoice = INV_FISHER;
s_ip = 1; s_iq = 1;
s_iStdErr = HESS;
```

and starting values

```
domega = 0;
vA = <0.10>';
vB = <0.89>';
dalpha = 0.18;
dkappa = 1;
```

After running the program (around 163 iterations) the output should say

```
Weak convergence (no improvement in line search) using numerical derivatives
Log likelihood value = -2120.918439
Parameters with standard errors:
omega    0.16571 (0.04744)
A1       0.13533 (0.01885)
B1       0.92677 (0.02108)
alpha   0.047146 (0.00081)
kappa     2584.1 (2.18176)
The value of the shape parameter kappa is estimated at a large value (>1000),
it is advised to apply the log-normal distribution to the dataset.
The Log-Normal distribution is a limiting case (kappa --> .Inf) of the
Generalised Gamma distribution.
Choose s_iDistribution == LOG_NORM to select the Log-Normal distribution.
```

The output window should be like the one showed in Figure 2. The model warns us for a large value of the shape parameter κ and advises the user to select the Log-Normal distribution since the Log-Normal distribution is a special case of the Generalised Gamma distribution for $\kappa \to \infty$, see Meeker and Escobar (1998).

Following the advice of the program we change the distribution to the Log-Normal distribution. For this we change the input block to

```
s_iDistribution = LOG_NORM;
s_iLinkFunction = LAMBDA;
s_iScalingChoice = INV_FISHER;
s_ip = 1; s_iq = 1;
s_iStdErr = HESS;
```

and starting values

```
domega = 0;
vA = <0.10>';
vB = <0.89>';
dalpha = 0.18;
```

After running the program (around 26 iterations) the output should say

```
Strong convergence using numerical derivatives
Log likelihood value = -2118.227620
Parameters with standard errors:
omega       0.16528 (0.04743)
A1          0.13539 (0.01883)
B1          0.92695 (0.02108)
alpha       0.17328 (0.00598)
```

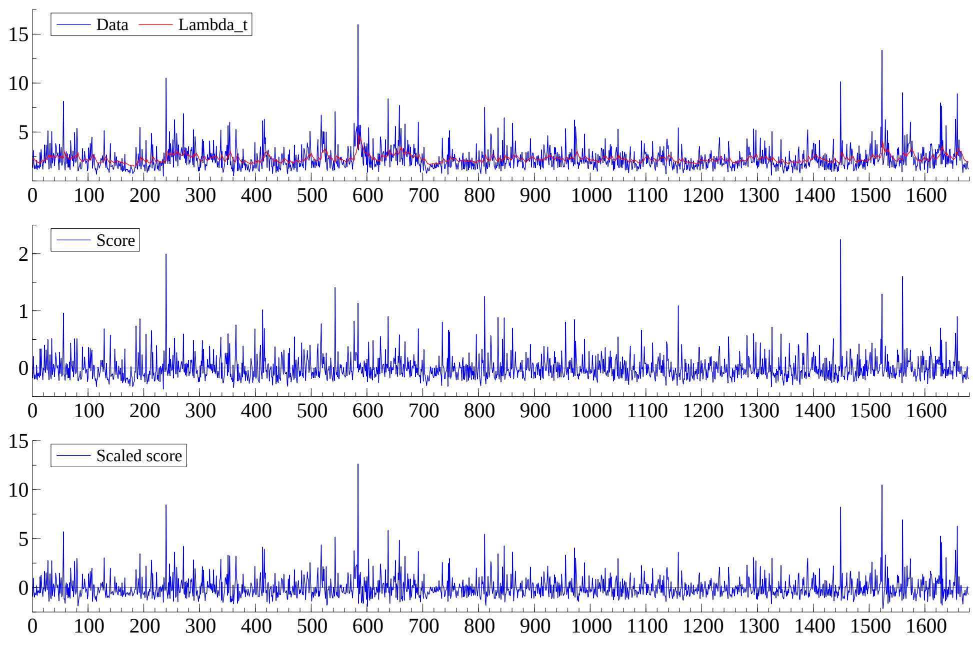
with the output window as showed in Figure 3.

To compare the estimated duration of two distributions in one figure, activate the code lines 304, 305 and 306 of the program. Do this by removing the two slashes in front of the lines. Also, place two slashes in front of line 301, 302 and 303. Set lines 304 and 305 to

```
ichoice1 = EXP;
ichoice2 = WEI;
```

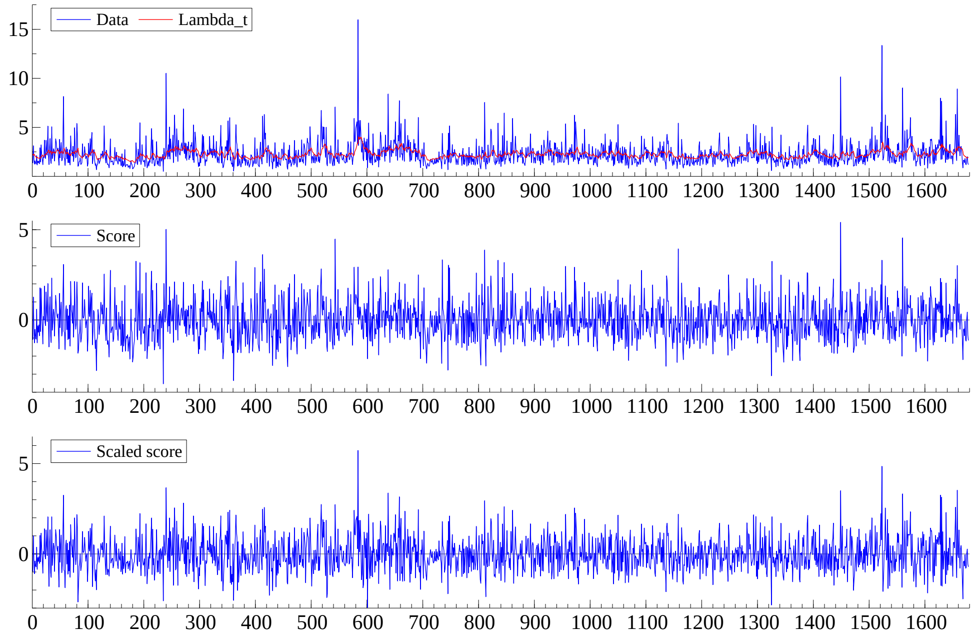
to compare the Exponential distribution with the Weibull distribution. After running the program, the output window should now look like Figure 4.

Figure 1: Exponential: estimated duration, score and scaled score



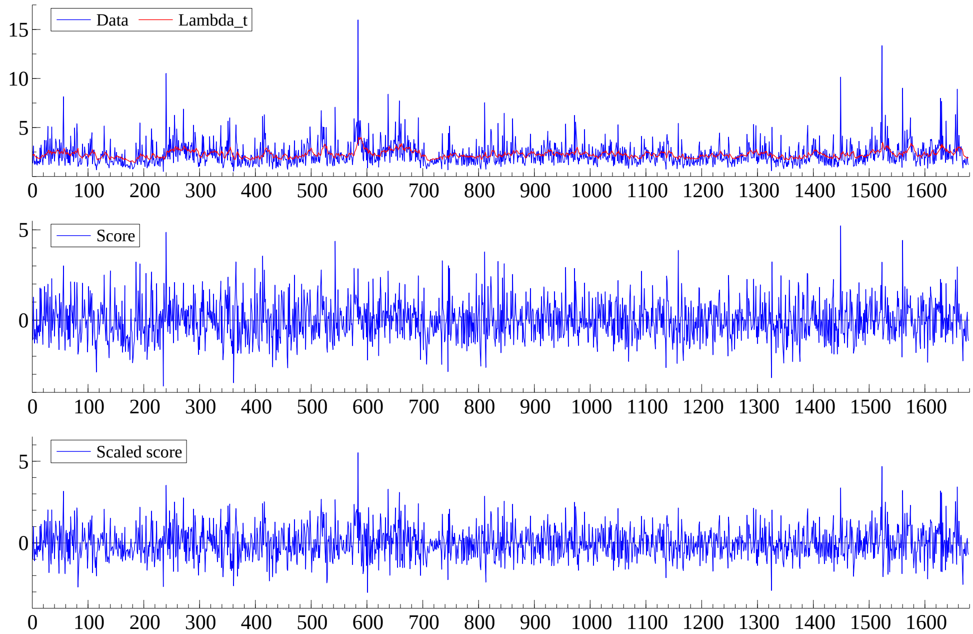
The top panel shows the adjusted intraday trading durations of 3M stock in December 1999 and the estimated duration. The mid panel shows the score and the bottom panel shows the scaled score.

Figure 2: Generalised Gamma: estimated duration, score and scaled score



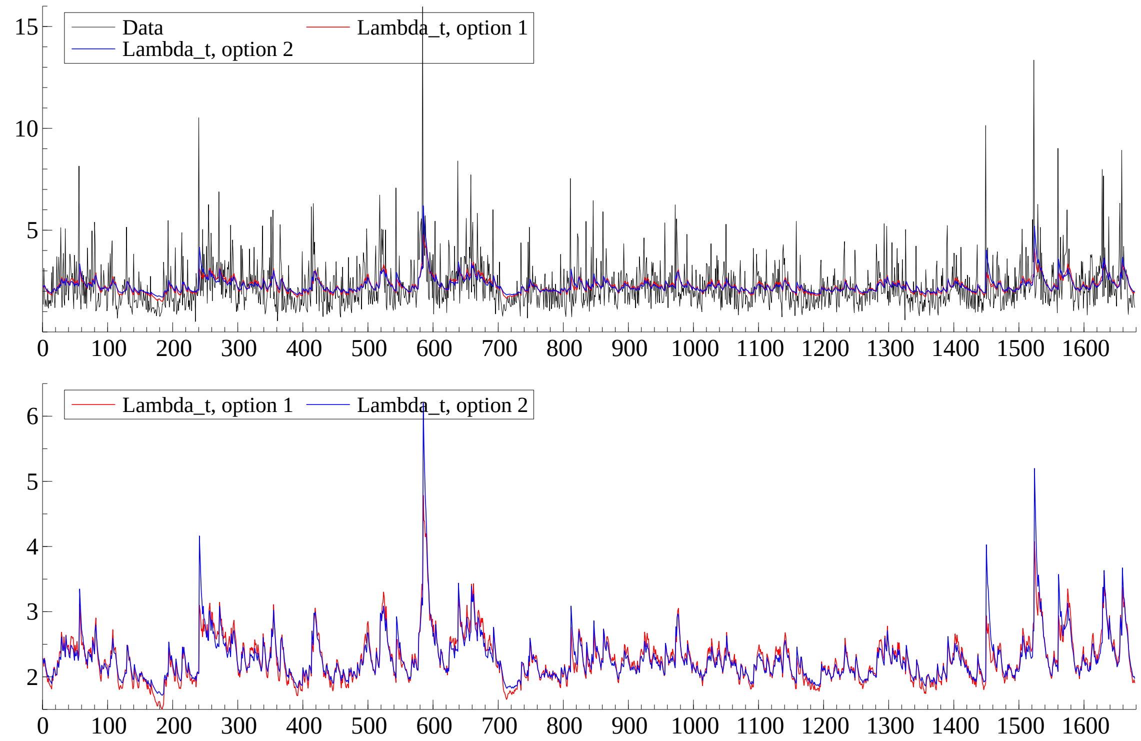
The top panel shows the adjusted intraday trading durations of 3M stock in December 1999 and the estimated duration. The mid panel shows the score and the bottom panel shows the scaled score.

Figure 3: Log-Normal: estimated duration, score and scaled score



The top panel shows the adjusted intraday trading durations of 3M stock in December 1999 and the estimated duration. The mid panel shows the score and the bottom panel shows the scaled score.

Figure 4: Comparison between Exponential and Weibull model



The top panel shows the adjusted intraday trading durations of 3M stock in December 1999 and the estimated duration for Exponential and Weibull GAS model. The bottom panel shows the same estimated duration only without the data.

References

Creal, D. D., S. J. Koopman, and A. Lucas (2013). Generalized autoregressive score models with applications. *J. Applied Econometrics 28*, forthcoming.

Meeker, Q. W. and L. A. Escobar (1998). *Statistical Methods for Reliability Data* (1st ed.). New York: Wiley-Interscience.

Tsay, R. S. (2005). *Analysis of financial time series* (2nd ed.). New Jersey: Wiley-Interscience.