

The Iterated Bootstrap

by

Russell Davidson

Department of Economics and CIREQ
McGill University
Montréal, Québec, Canada
H3A 2T7

GREQAM
Centre de la Vieille Charité
2 Rue de la Charité
13236 Marseille cedex 02, France

russell.davidson@mcgill.ca

and

Mirza Trokić

Department of Economics
McGill University
Montréal, Québec, Canada
H3A 2T7

mirza.trokić@mail.mcgill.ca

Abstract

The standard forms of bootstrap iteration are very computationally demanding. As a result, there have been several attempts to alleviate the computational burden by use of approximations. In this paper, we extend the fast double bootstrap of Davidson and MacKinnon (2007) to higher orders of iteration, and provide algorithms for their implementation. The new methods make computational demands that increase only linearly with the level of iteration, unlike standard procedures, whose demands increase exponentially. In a series of simulation experiments, we show that the fast triple bootstrap improves on both the standard and fast double bootstraps, in the sense that it suffers from less size distortion under the null with no accompanying loss of power.

Key words: bootstrap iteration, fast iterated bootstrap

JEL codes: C10, C12, C15, C63

This research was supported by the Canada Research Chair program (Chair in Economics, McGill University), and by grants from the Social Sciences and Humanities Research Council of Canada and the Fonds Québécois de Recherche sur la Société et la Culture.

August 2011

1. Introduction

Bootstrap iteration has been discussed in a wide variety of contexts from not long after the invention of the bootstrap by Efron (1979). The first article to mention the topic of the iterated bootstrap to our knowledge is Hall (1986), a paper followed quickly by two articles by Beran (1987) and (1988), in which the double bootstrap is introduced. An extensive theoretical discussion, with examples, is given in Hall and Martin (1988). Lee and Young (1995) develop procedures for bootstrap confidence intervals with an iterated bootstrap that does not involve any resampling, replacing it with analytic asymptotic approximations. In Lee and Young (2003), they investigate the possibilities of weighted resampling to mimic the results of bootstrap iteration. Ho and Lee (2005) consider bootstrap iteration in conjunction with smoothing of the discrete distribution associated with conventional resampling.

Much of the work cited above is motivated by a desire to alleviate the crushing computational burden of bootstrap iteration. Another attempt along those lines is the fast double bootstrap (FDB) exposted in Davidson and MacKinnon (2007). The technique had been used previously: Davidson and MacKinnon (2002) use it to improve the reliability of bootstrap tests of non-nested linear regression models; Omtzigt and Fachin in a (2002) working paper show that it gives better size correction than the single bootstrap with or without Bartlett correction in the cointegrated VAR model; Lamarche (2004) investigates some of its numerical properties; Davidson (2006) uses it in developing bootstrap tests of cointegration with fractionally integrated time series. A recent working paper, Ouyse (2009) develops a version of the FDB for the purposes of bias correction.

The starting point for this paper is found in Davidson (2010), in which the FDB is studied and compared with the standard double bootstrap and some other procedures aimed at improving bootstrap reliability. In the context of a much simplified unit root test, it is shown that estimating the distribution of the FDB P value leads to a reduction in the size distortion of the FDB, which is less distorted than the single bootstrap, and roughly comparable to the standard double bootstrap. The procedure is more or less a bootstrap of the FDB, which gives rise to an approximation to the bootstrapped double bootstrap, that is, the twice iterated, or triple, bootstrap. The FDB algorithm is not a nested bootstrap procedure. But when it is bootstrapped, what results is a nested bootstrap procedure, essentially as computationally intensive as the double bootstrap.

It is therefore interesting to see whether it is possible to make use of the approximations that simplify the original double bootstrap algorithm to the FDB, but to the bootstrapped FDB. One would then effectively obtain an approximation to the standard triple bootstrap, an approximation that can reasonably be termed the fast triple bootstrap (FTB). In this paper, after having discussed bootstrap iteration and developing some suitable notation, we show how fast versions of any order of iterated bootstrap can be defined.

In the next section, we set up notation for dealing with bootstrap iteration, and give an algorithm for implementing the standard double bootstrap, based on an algorithm for estimating the rejection probability (RP) of a conventional single bootstrap test. Then, in section 3, we review the theory of the FDB, linking it to a fast algorithm for estimating

the RP of the single bootstrap. Section 4 is where we pursue an approach analogous to that which leads to the FDB in order to motivate and define the fast triple bootstrap, for which a computational algorithm is detailed. Once we can see how to get as far as the FTB, the road is clear to defining arbitrary orders of fast iterated bootstraps; that is done in section 5. In section 6, we report the results of three sets of simulation experiments, one based on the experimental design of Davidson (2010) for a unit root test, the next on a test for ARCH effects, and the third on a test for serial correlation of the disturbances of a regression model. Section 7 concludes.

2. Concepts and Notations

We denote by \mathbb{M}_0 the set of data-generating processes (DGPs) that represent a null hypothesis we wish to test. The test statistic used is denoted by τ . In general, τ has a different distribution under the different DGPs in \mathbb{M}_0 , and it certainly has a different distribution under DGPs in the model, \mathbb{M} say, that represents the alternative hypothesis. Here $\mathbb{M}_0 \subset \mathbb{M}$. It is conventional to suppose that τ is defined as a random variable on some suitable probability space, on which we define a different probability measure for each different DGP.

Rather than using this approach, we define a probability space (Ω, \mathcal{F}, P) , with just one probability measure, P . Then we treat the test statistic τ as a stochastic process with as index set the set \mathbb{M} . We have

$$\tau : \mathbb{M} \times \Omega \rightarrow \mathbb{R}.$$

Since most of the discussion of the paper is couched in the language of simulation, the probability space can, for our present purposes, be taken to be that of a random number generator. A realisation of the test statistic is therefore written as $\tau(\mu, \omega)$, for some $\mu \in \mathbb{M}$ and $\omega \in \Omega$. Throughout the following discussion, we suppose that, under any DGP μ that we may consider, the distribution of the random variable $\tau(\mu, \cdot)$ is absolutely continuous with respect to Lebesgue measure on \mathbb{R} .

For notational convenience, we suppose that the range of τ is the $[0, 1]$ interval rather than the whole real line, and that the statistic takes the form of an approximate P value, which thus leads to rejection when the statistic is too small. If τ is a pivotal statistic with respect to \mathbb{M}_0 , then a realisation $\tau(\mu, \omega)$ depends only on ω for any argument $\mu \in \mathbb{M}_0$. In such a case, inference can be based on an (exact) Monte Carlo test. Let $R_0 : [0, 1] \rightarrow [0, 1]$ be the CDF of τ under any DGP $\mu \in \mathbb{M}_0$:

$$R_0(\alpha) = P\{\omega \in \Omega \mid \tau(\mu, \omega) \leq \alpha\}. \quad (1)$$

Suppose that we have a statistic computed from a data set that may or may not have been generated by a DGP in \mathbb{M}_0 . Denote this statistic as t . Then the ideal P value that would give exact inference is $R_0(t)$. If t is generated by a DGP in \mathbb{M}_0 , $R_0(t)$ is distributed as $U(0,1)$, but not, in general, if t comes from some other DGP.

For the Monte Carlo test, R_0 is estimated by simulation. One generates B “bootstrap” statistics $\tau(\mu, \omega_j^*)$, $j = 1, \dots, B$ where μ is any suitable DGP in \mathbb{M}_0 , and the ω_j^* are

independent. Each ω_j^* can be thought of as a set of those random numbers needed to generate a realisation of the statistic. Then the P value on which inference is based is the proportion of the B bootstrap statistics that are more extreme than t . We can therefore write

$$\hat{R}_0(t) = \frac{1}{B} \sum_{j=1}^B \mathbf{I}(\tau(\mu, \omega_j^*) < t), \quad (2)$$

where \mathbf{I} is an indicator function, and $\mu \in \mathbb{M}_0$. As $B \rightarrow \infty$, $\hat{R}_0(t)$ tends almost surely to $R_0(t)$, which is a deterministic function of t .

Suppose now that τ is not pivotal with respect to \mathbb{M}_0 . In that case, exact inference is no longer possible. We must now index the CDF R_0 by the DGP μ . The definition (1) still applies, but the left-hand side is now written as $R_0(\alpha, \mu)$. If the DGP that generated t , μ_0 say, does belong to \mathbb{M}_0 , then $R_0(t, \mu_0)$ is $U(0,1)$. But this fact cannot be used for inference, since μ_0 is unknown.

The principle of the bootstrap is that, when we want to use some function or functional of an unknown DGP μ_0 , we use an estimate in its place. This estimate is the bootstrap DGP, which we denote by β . Then the bootstrap statistic that follows the $U(0,1)$ distribution approximately is $R_0(t, \beta)$. Analogously to (2), we make the definition

$$\hat{R}_0(\alpha, \mu) = \frac{1}{B} \sum_{j=1}^B \mathbf{I}(\tau(\mu, \omega_j^*) < \alpha), \quad (3)$$

the ω_j^* as before. Then, as $B \rightarrow \infty$, $\hat{R}_0(\alpha, \mu)$ tends almost surely to $R_0(\alpha, \mu)$. Accordingly, we estimate the bootstrap statistic by $\hat{R}_0(t, \beta)$.

Just as t is $\tau(\mu_0, \omega)$ where neither the true DGP μ_0 nor the realisation ω is observed, so also the bootstrap DGP β can be expressed as $b(\mu_0, \omega)$, for the same unobserved μ_0 and ω as for t . We have

$$b : \mathbb{M} \times \Omega \rightarrow \mathbb{M}_0,$$

where, although the model \mathbb{M} on the left-hand side may correspond to the entire alternative hypothesis, we insist that the \mathbb{M}_0 on the right-hand side is the null model under test (the first golden rule of bootstrapping). Note here that, since the distribution of τ is absolutely continuous for all $\mu \in \mathbb{M}_0$, its distribution under the bootstrap DGP β is also absolutely continuous. With this definition, the bootstrap statistic $R_0(t, \beta)$ can be written as $p_1(\mu_0, \omega)$, where the new function $p_1 : \mathbb{M} \times \Omega \rightarrow [0, 1]$ is defined as follows:

$$p_1(\mu, \omega) \equiv R_0(\tau(\mu, \omega), b(\mu, \omega)). \quad (4)$$

Since by absolute continuity R_0 is a continuous function, it follows that p_1 also has an absolutely continuous distribution. We denote the continuous CDF of $p_1(\mu, \omega)$ by $R_1(\cdot, \mu)$.

Consider now a simulation experiment designed to estimate the function $R_1(\cdot, \mu)$ for some given μ . Ignoring for the moment any tricks of variance reduction that we might use in particular cases, we can design the experiment as follows.

Algorithm R1:

1. For each $i = 1, \dots, N$:
 - (i) Generate an independent realisation ω_i from (Ω, \mathcal{F}, P) ;
 - (ii) Compute a statistic $t_i \equiv \tau(\mu, \omega_i)$ and the corresponding bootstrap DGP $\beta_i \equiv b(\mu, \omega_i)$;
 - (iii) Compute the estimate $\hat{R}_0(t_i, \beta_i)$ of the bootstrap statistic.
2. Then for any $\alpha \in [0, 1]$, the estimate of $R_1(\alpha, \mu)$ is

$$\hat{R}_1(\alpha, \mu) \equiv \frac{1}{N} \sum_{i=1}^N \mathbf{I}(\hat{R}_0(t_i, \beta_i) < \alpha). \quad (5)$$

Of course for each i , step (iii) involves generating B statistics that take the form $\tau(\beta_i, \omega_{ij}^*)$, $j = 1, \dots, B$.

The random variable $R_1(p_1(\mu, \omega), \mu)$ is, by construction, distributed as $U(0,1)$. But, as with $R_0(\tau(\mu, \omega), \mu)$, which is also distributed as $U(0,1)$, this fact is not enough to allow exact inference, because the actual μ_0 that generates the data is unknown outside the context of a simulation experiment.

However, the bootstrap principle can again be applied, and the unknown μ_0 replaced by an estimate $\beta = b(\mu_0, \omega)$, which is directly observed, or at least can be calculated from observed data. This leads to the double bootstrap, of which the P value, for realisations t and β , can be written as

$$\hat{R}_1(\hat{R}_0(t, \beta), \beta).$$

Under the assumption that the true μ_0 belongs to the null model \mathbb{M}_0 , this is an estimate of the probability mass in the distribution of the single bootstrap statistic to the left of the estimate $\hat{R}_0(t, \beta)$ of $R_0(t, \beta)$. As both B and N tend to infinity, the double bootstrap P value tends almost surely to $R_1(R_0(t, \beta), \beta)$. Expressed as a random variable, this limiting P value is

$$p_2(\mu, \omega) \equiv R_1(R_0(\tau(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)) \quad (6)$$

evaluated at $\mu = \mu_0$. If we write the right-hand side above as $R_1(p_1(\mu, \omega), b(\mu, \omega))$, the analogy with the definition (4) of $p_1(\mu, \omega)$ is complete. This demonstrates that the double bootstrap, by estimating the probability mass to the left of the single bootstrap P value, effectively bootstraps the single bootstrap P value.

From that observation, it is clear that we can define iterated bootstraps as follows. For $k = 0, 1, 2, \dots$, we define

$$R_k(\alpha, \mu) = P\{\omega \in \Omega \mid p_k(\mu, \omega) \leq \alpha\}, \quad (7)$$

$$p_{k+1}(\mu, \omega) = R_k(p_k(\mu, \omega), b(\mu, \omega)), \quad (8)$$

where we initialise the recursion by the definition $p_0(\mu, \omega) = \tau(\mu, \omega)$. Thus $p_{k+1}(\mu, \omega)$ is the bootstrap P value obtained by bootstrapping the k^{th} order P value $p_k(\mu, \omega)$. It estimates the probability mass in the distribution of the k^{th} order P value to the left of its realisation.

Since $R_2(\cdot, \mu)$ is the CDF under μ of $p_2(\mu, \omega)$, we could estimate $R_2(\alpha, \mu)$ by the expression

$$\frac{1}{B} \sum_{j_2=1}^B \mathbf{I}(p_2(\mu, \omega_{j_2}^*) < \alpha)$$

if we could compute $p_2(\mu, \omega)$ exactly. In practice, we must estimate it by simulation. Since by (8) we have $p_2(\mu, \omega) = R_1(p_1(\mu, \omega), b(\mu, \omega))$, a suitable estimate is

$$\hat{p}_2(\mu, \omega) \equiv \hat{R}_1(\hat{p}_1(\mu, \omega), b(\mu, \omega)) = \hat{R}_1(\hat{R}_0(p_0(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)). \quad (9)$$

This allows us to define the function \hat{R}_2 by the formula

$$\hat{R}_2(\alpha, \mu) = \frac{1}{B} \sum_{j=1}^B \mathbf{I}(\hat{p}_2(\mu, \omega_j^*) < \alpha).$$

Going on, we have

$$\begin{aligned} p_3(\mu, \omega) &= R_2(p_2(\mu, \omega), b(\mu, \omega)) = R_2(R_1(p_1(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)) \\ &= R_2(R_1(R_0(p_0(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)). \end{aligned} \quad (10)$$

This can be estimated by $\hat{p}_3(\mu, \omega) \equiv \hat{R}_2(\hat{R}_1(\hat{R}_0(p_0(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega))$, or by $\hat{R}_2(\hat{p}_2(\mu, \omega), b(\mu, \omega))$, by (9). We are therefore led to the following definitions, analogous to (7) and (8), for $k = 0, 1, 2, \dots$:

$$\hat{R}_k(\alpha, \mu) = \frac{1}{B} \sum_{j=1}^B \mathbf{I}(\hat{p}_k(\mu, \omega_{j_k}^*) < \alpha), \quad (11)$$

$$\hat{p}_{k+1}(\mu, \omega) = \hat{R}_k(\hat{p}_k(\mu, \omega), b(\mu, \omega)). \quad (12)$$

Note that, since $p_0(\mu, \omega) = \tau(\mu, \omega)$ is supposed to be observed directly, we can put $\hat{p}_0 = p_0$.

The above definitions allow us to express iterated bootstrap P values and their estimates as functions of the DGP μ and the realisation ω . But in fact they depend on μ and ω only through $t = \tau(\mu, \omega)$ and $\beta = b(\mu, \omega)$. We have seen that the first-level P value is estimated by $\hat{R}_0(t, \beta)$, the second-level one by $\hat{R}_1(\hat{R}_0(t, \beta), \beta)$, and it is easy to see from (10) that the third-level one is estimated by $\hat{R}_2(\hat{R}_1(\hat{R}_0(t, \beta), \beta), \beta)$. Thus, expressing the estimated P values as functions of t and β alone, (12) becomes

$$\hat{p}_{k+1}(t, \beta) = \hat{R}_k(\hat{p}_k(t, \beta), \beta).$$

The function \hat{R}_k is still defined by (11) with no change.

Our notations have been carefully chosen in order to avoid needless complexity. Nevertheless, they conceal considerable detail that has to be taken into account for the purposes of computation. The second-level estimated P value is $\hat{R}_1(\hat{R}_0(t, \beta), \beta)$, which expands to

$$\frac{1}{B} \sum_{j=1}^B \mathbb{I} \left[\frac{1}{B_1} \sum_{j_1=1}^{B_1} \mathbb{I} \left(\tau(b(\beta, \omega_j^*), \omega_{jj_1}^*) < \tau(\beta, \omega_j^*) \right) < \hat{p}_1(t, \beta) \right], \quad (13)$$

where

$$\hat{p}_1(t, \beta) = \frac{1}{B} \sum_{j=1}^B \mathbb{I}(\tau(\beta, \omega_j^*) < t). \quad (14)$$

While the value of B in (14) may be equal to that in (13), B_1 should be relatively prime to B , so as to avoid ties in the first inequality in (13). Comparable expressions for the estimated triple bootstrap P value are left to the reader's imagination.

It may be easier and more adapted to computation to express (13) as an algorithm. We have:

Algorithm P2:

1. From the data set under analysis, compute the realised statistic t and the bootstrap DGP β . These are taken to be the realisations $\tau(\mu, \omega)$ and $b(\mu, \omega)$ respectively, where μ may or may not belong to the null model \mathbb{M}_0 .
2. Using the DGP β , draw B bootstrap samples and compute the B bootstrap test statistics $\tau(\beta, \omega_j^*)$, $j = 1, \dots, B$, and the B iterated bootstrap DGPs $b(\beta, \omega_j^*)$ in exactly the same way as t and β were computed.
3. Compute the estimated first-level bootstrap P value $\hat{p}_1(t, \beta)$, using (14) above.
4. Then, for each $j = 1, \dots, B$, draw B_1 iterated bootstrap samples from $b(\beta, \omega_j^*)$, and compute the B_1 second-level statistics $\tau(b(\beta, \omega_j^*), \omega_{jj_1}^*)$, $j_1 = 1, \dots, B_1$.
5. Compute the j^{th} iterated bootstrap P value as:

$$p_j^* = \frac{1}{B_1} \sum_{j_1=1}^{B_1} \mathbb{I}(\tau(b(\beta, \omega_j^*), \omega_{jj_1}^*) < \tau(\beta, \omega_j^*)).$$

6. Finally, obtain the estimated double bootstrap P value as:

$$\hat{p}_2(t, \beta) = \frac{1}{B} \sum_{j=1}^B \mathbb{I}(p_j^* < \hat{p}_1(t, \beta)).$$

3. The Fast Double Bootstrap

The fast double bootstrap (FDB) of Davidson and MacKinnon (2007) is based on two approximations. The first is to assume that, for any $\mu \in \mathbb{M}_0$, the random variables $\tau(\mu, \omega)$ and $b(\mu, \omega)$ are independent. The assumption is of course false except in special circumstances, but it holds asymptotically in many commonly encountered situations. By definition,

$$R_1(\alpha, \mu) = P\{\omega \in \Omega \mid p_1(\mu, \omega) < \alpha\} = E[\mathbf{I}(R_0(\tau(\mu, \omega), b(\mu, \omega)) < \alpha)]. \quad (15)$$

Let $Q_0(\cdot, \mu)$ be the quantile function corresponding to the distribution $R_0(\cdot, \mu)$. Since R_0 is absolutely continuous, we have

$$R_0(Q_0(\alpha, \mu), \mu) = \alpha = Q_0(R_0(\alpha, \mu), \mu).$$

Use of this relation between R_0 and Q_0 lets us write (15) as

$$R_1(\alpha, \mu) = E[\mathbf{I}(\tau(\mu, \omega) < Q_0(\alpha, b(\mu, \omega)))]$$

If $\tau(\mu, \omega)$ and $b(\mu, \omega)$ are treated as though they were independent, then we have

$$\begin{aligned} R_1(\alpha, \mu) &= E\left[E[\mathbf{I}(\tau(\mu, \omega) < Q_0(\alpha, b(\mu, \omega))) \mid b(\mu, \omega)]\right] \\ &= E[R_0(Q_0(\alpha, b(\mu, \omega)), \mu)] \end{aligned} \quad (16)$$

where the last step follows from the Independence Lemma. Since in general $\tau(\mu, \omega)$ and $b(\mu, \omega)$ are not independent, (16) is taken as an approximation.

Consider now two identical probability spaces $(\Omega_1, \mathcal{F}_1, P_1)$ and $(\Omega_2, \mathcal{F}_2, P_2)$, and their product space $(\Omega_1 \times \Omega_2, \mathcal{F}_1 \times \mathcal{F}_2, P_1 \times P_2)$. Define the stochastic process

$$\tau^1 : \mathbb{M} \times (\Omega_1 \times \Omega_2) \rightarrow \mathbb{R}$$

by the formula

$$\tau^1(\mu, \omega_1, \omega_2) = \tau(b(\mu, \omega_1), \omega_2). \quad (17)$$

Thus $\tau^1(\mu, \omega_1, \omega_2)$ can be thought of as a realisation of the bootstrap statistic when the underlying DGP is μ . We denote the CDF of τ^1 under μ by $R^1(\cdot, \mu)$. Thus

$$\begin{aligned} R^1(\alpha, \mu) &= (P \times P^*)\{(\omega_1, \omega_2) \in \Omega_1 \times \Omega_2 \mid \tau(b(\mu, \omega_1), \omega_2) < \alpha\} \\ &= E[\mathbf{I}(\tau(b(\mu, \omega_1), \omega_2) < \alpha)] \\ &= E\left[E[\mathbf{I}(\tau(b(\mu, \omega_1), \omega_2) < \alpha) \mid \mathcal{F}_1]\right] \\ &= E[R_0(\alpha, b(\mu, \omega_1))], \end{aligned} \quad (18)$$

where the last step follows from the Independence Lemma.

The second approximation underlying the FDB can now be stated as follows:

$$\mathbb{E}[R_0(Q_0(\alpha, b(\mu, \omega)), \mu)] \approx R_0(Q^1(\alpha, \mu), \mu), \quad (19)$$

where $Q^1(\cdot, \mu)$ is the quantile function inverse to the CDF $R^1(\cdot, \mu)$. Since by definition $R^1(Q^1(\alpha, \mu), \mu) = \alpha$, it follows from (18) that

$$\mathbb{E}[R_0(Q^1(\alpha, \mu), b(\mu, \omega))] = \alpha. \quad (20)$$

In order to motivate the approximation (19), we follow Davidson and MacKinnon (2007), and suppose that, for any DGP $\mu \in \mathbb{M}_0$ and all $\alpha \in [0, 1]$, $R_0(\alpha, \mu) - \alpha$ is small in some appropriate sense. In other words, suppose that τ is approximately pivotal with respect to \mathbb{M}_0 . Next, assume that R_0 is not only continuous but also continuously differentiable with respect to its first argument α for all $\mu \in \mathbb{M}_0$. Thus the statistic τ has a continuous density for all $\mu \in \mathbb{M}_0$. Finally, we assume that $R'_0(\alpha, \mu) - 1$, where R'_0 denotes the derivative of R_0 with respect to its first argument, is small in the same sense as that in which $R_0(\alpha, \mu) - \alpha$ is small.

The assumption about the derivative R'_0 implies that $Q_0(\alpha, \mu) - \alpha$ is small for $\mu \in \mathbb{M}_0$. The definition (18) implies that $R^1(\alpha, \mu) - \alpha$ is small, and so also $Q^1(\alpha, \mu) - \alpha$. Now (20) can be written as

$$\mathbb{E}[R_0(Q^1(\alpha, \mu), b(\mu, \omega)) - R_0(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega))] = 0,$$

and our assumption about the derivative of R_0 , along with Taylor's Theorem, lets us rewrite this equation as

$$\mathbb{E}[(1 + \eta_1)(Q^1(\alpha, \mu) - Q_0(\alpha, b(\mu, \omega)))] = 0, \quad (21)$$

where the random variable η_1 is small. Further applications of our smallness assumptions give us

$$Q^1(\alpha, \mu) - Q_0(\alpha, b(\mu, \omega)) = \alpha - \alpha + \eta_2$$

where η_2 is another small random variable. Thus (21) becomes

$$\mathbb{E}[Q_0(\alpha, b(\mu, \omega))] = Q^1(\alpha, \mu) + \mathbb{E}(\eta_1 \eta_2). \quad (22)$$

Thus the expectation of $Q_0(\alpha, b(\mu, \omega))$ is equal to $Q^1(\alpha, \mu)$ up to an error of the second order of small quantities.

The difference between the left- and right-hand sides of (19) is

$$\begin{aligned} & \mathbb{E}[R_0(Q_0(\alpha, b(\mu, \omega)), \mu) - R_0(Q^1(\alpha, \mu), \mu)] \\ &= \mathbb{E}[(1 + \eta_3)(Q_0(\alpha, b(\mu, \omega)) - Q^1(\alpha, \mu))], \end{aligned}$$

where η_3 is small. By (22) the last expression above is a sum of products of two small quantities, thus justifying the approximation (19).

On putting the two approximations, (16) and (19), together, we obtain

$$R_1(\alpha, \mu) \approx R_0(Q^1(\alpha, \mu), \mu) \equiv R_1^f(\alpha, \mu). \quad (23)$$

The approximation $R_1^f(\alpha, \mu)$ can be estimated by simulation as follows.

Algorithm FastR1:

1. For each $i = 1, \dots, N$:
 - (i) Generate an independent realisation ω_i from (Ω, \mathcal{F}, P) ;
 - (ii) Compute a statistic $t_i = \tau(\mu, \omega_i)$ and the corresponding bootstrap DGP $\beta_i = b(\mu, \omega_i)$;
 - (iii) Generate a second independent realisation ω_i^* , and a realisation $t_i^1 = \tau(\beta_i, \omega_i^*)$ of the random variable τ^1 .
2. Sort the t_i^1 in increasing order, and form an estimate $\hat{Q}^1(\alpha, \mu)$ of $Q^1(\alpha, \mu)$ as the $[\alpha N]$ order statistic. (Here it is advantageous to have $\alpha(N + 1)$ an integer.)
3. Estimate $R_0(Q^1(\alpha, \mu), \mu)$ by the proportion of the t_i less than the estimate of $Q^1(\alpha, \mu)$.

Denote the estimate obtained by the above algorithm as $\hat{R}_1^f(\alpha, \mu)$. The algorithm is obviously much faster to implement than [Algorithm R1](#), since there is no inner loop in step (iii).

The theoretical double bootstrap P value is the random variable $p_2(\mu, \omega)$ of (6). If we use the approximation (23) for the function R_1 , we obtain an expression for the theoretical fast double bootstrap P value:

$$p_2^f(\mu, \omega) \equiv R_0(Q^1(R_0(\tau(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)). \quad (24)$$

A simulation-based estimate of this is

$$\hat{p}_2^f(t, \beta) = \hat{R}_0(\hat{Q}^1(\hat{R}_0(t, \beta), \beta), \beta) = \hat{R}_1^f(\hat{p}_1(t, \beta), \beta). \quad (25)$$

As before, it is better adapted to computation to express this estimate as an algorithm.

Algorithm FDB:

1. From the data set under analysis, compute the realised statistic t and the bootstrap DGP β .
2. Draw B bootstrap samples and compute B bootstrap statistics $t_j^* = \tau(\beta, \omega_j^*)$, $j = 1, \dots, B$, and B iterated bootstrap DGPs $\beta_j^* = b(\beta, \omega_j^*)$.
3. Compute B second-level bootstrap statistics $t_j^{1*} = \tau(\beta_j^*, \omega_j^{**})$, and sort them in increasing order.
4. Compute the estimated first-level bootstrap P value $\hat{p}_1(t, \beta)$ as the proportion of the t_j^* smaller than t .
5. Obtain the estimate $\hat{Q}^1(\hat{p}_1(t, \beta), \beta)$ as the order statistic of the t_j^{1*} of rank $[B\hat{p}_1(t, \beta)]$. Denote this estimate as \hat{Q}^{1*} .
6. The estimated FDB P value is the proportion of the t_j^* that are smaller than \hat{Q}^{1*} .

4. The Fast Triple Bootstrap

In order to study the distribution of the FDB P value (24), we wish to evaluate the expression

$$\mathbb{E}\left[\mathbf{I}\left(R_0(Q^1(R_0(\tau(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)) < \alpha)\right)\right],$$

which is the probability, under the DGP μ , that the FDB P value is less than α . The inequality that is the argument of the indicator above is equivalent to several other inequalities, as follows:

$$\begin{aligned} & Q^1(R_0(\tau(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)) < Q_0(\alpha, b(\mu, \omega)) \\ \iff & R_0(\tau(\mu, \omega), b(\mu, \omega)) < R^1(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega)) \\ \iff & \tau(\mu, \omega) < Q_0(R^1(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)). \end{aligned}$$

At this point, we can again invoke an approximation that would be exact if $\tau(\mu, \omega)$ and $b(\mu, \omega)$ were independent. The final inequality above separates $\tau(\mu, \omega)$ from $b(\mu, \omega)$ on the left- and right-hand sides respectively, and so the expectation of the indicator of that inequality is approximated by

$$\mathbb{E}\left[R_0(Q_0(R^1(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)), \mu)\right]. \quad (26)$$

It is quite possible to estimate (26) for given α and μ by simulation. The function $R_0(\cdot, \mu)$ can be estimated as in (3). But, for the other needed functions, $Q_0(\cdot, b(\mu, \omega))$ and $R^1(\cdot, b(\mu, \omega))$, which both depend on ω , a second, inner, loop is necessary for each iteration of the main simulation loop. Thus the straightforward estimation procedure is about as computationally intensive as the double bootstrap.

However, we can make a further approximation in the spirit of (19), the second of the approximations that lead to the FDB. The aim is to use our smallness assumptions in order to eliminate all mention of the bootstrap DGP $b(\mu, \omega)$, the presence of which requires the inner loop.

Since the derivatives of R_0 , Q_0 , and R^1 with respect to their first argument are close to 1, it follows that the derivative with respect to α of $R_0(Q_0(R^1(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)), \mu)$ is also equal to 1 plus a small quantity. Therefore, by Taylor's Theorem,

$$\begin{aligned} & R_0(Q_0(R^1(Q_0(\alpha, b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)), \mu) - R_0(Q_0(R^1(Q^1(\alpha, \mu), b(\mu, \omega)), b(\mu, \omega)), \mu) \\ &= (1 + \eta)(Q_0(\alpha, b(\mu, \omega)) - Q^1(\alpha, \mu)), \end{aligned}$$

η small. The expectation of the right-hand side above is a small quantity of the second order, and so the expectation (26) is, up to an error of that order,

$$\mathbb{E}\left[R_0(Q_0(R^1(Q^1(\alpha, \mu), b(\mu, \omega)), b(\mu, \omega)), \mu)\right]. \quad (27)$$

Analogously to (17), define the random variable

$$\tau^2(\mu, \omega_1, \omega_2, \omega_3) = \tau(b(b(\mu, \omega_1), \omega_2), \omega_3),$$

which can be thought of as a realisation of the second-order bootstrap statistic. The CDF of τ^2 under μ , denoted by $R^2(\cdot, \mu)$ is given by

$$\begin{aligned} R^2(\alpha, \mu) &= \mathbb{E}[\mathbb{I}(\tau^2(\mu, \omega_1, \omega_2, \omega_3) < \alpha)] \\ &= \mathbb{E}[\mathbb{E}[\mathbb{I}(\tau(b(b(\mu, \omega_1), \omega_2), \omega_3) < \alpha) \mid \mathcal{F}_{12})]] \\ &= \mathbb{E}[R_0(\alpha, b(b(\mu, \omega_1), \omega_2))] \\ &= \mathbb{E}[R^1(\alpha, b(\mu, \omega_1))], \end{aligned} \tag{28}$$

where \mathcal{F}_{12} denotes the product sigma-algebra defined on the probability space of ω_1 and ω_2 . The third equality follows from the Independence Lemma and the definition of R_0 , the fourth from the relation (16).

Now, an argument just like that leading to (27), but based on (28), shows that the expectation (27) is equal to

$$\mathbb{E}[R_0(Q_0(R^2(Q^1(\alpha, \mu), \mu), b(\mu, \omega)), \mu)]$$

up to an error small of second order. Finally, we can use the result (22) to show that this last expression is, always up to an error small of second order,

$$R_2^f(\alpha, \mu) \equiv R_0(Q^1(R^2(Q^1(\alpha, \mu), \mu), \mu), \mu). \tag{29}$$

Estimation of this by simulation, for given α and μ , can be done using the following algorithm.

Algorithm FastR2:

1. For each $i = 1, \dots, N$:
 - (i) Generate an independent realisation ω_{i1} from (Ω, \mathcal{F}, P) ;
 - (ii) Compute a statistic $t_i = \tau(\mu, \omega_{i1})$ and corresponding bootstrap DGP $\beta_{i1} = b(\mu, \omega_{i1})$;
 - (iii) Generate a second independent realisation ω_{i2} , a realisation $t_i^1 = \tau(\beta_{i1}, \omega_{i2})$ of τ^1 , and corresponding bootstrap DGP $\beta_{i2} = b(\beta_{i1}, \omega_{i2})$;
 - (iv) Generate a third independent realisation ω_{i3} and a realisation $t_i^2 = \tau(\beta_{i2}, \omega_{i3})$ of τ^2 .
2. Sort the t_i^1 in increasing order, and form an estimate $\hat{Q}^1(\alpha, \mu)$ as the order statistic of rank $\lceil \alpha N \rceil$.
3. Estimate $R^2(Q^1(\alpha, \mu), \mu)$ by the proportion of the t_i^2 that are less than $\hat{Q}^1(\alpha, \mu)$. Denote the estimate by \hat{r}_2 .

4. Estimate $Q^1(R^2(Q^1(\alpha, \mu), \mu), \mu)$ as the order statistic of the t_i^1 of rank $\lceil \hat{r}_2 N \rceil$. Denote the estimate by \hat{q}_1 .
5. Finally, estimate (29) as the proportion of the t_i that are smaller than \hat{q}_1 .

The theoretical FDB P value (24) is the approximation (23) evaluated with α set equal to the first-level bootstrap P value, and μ replaced by the bootstrap DGP. The theoretical fast triple bootstrap (FTB) P value is formed analogously from (29) by setting α equal to the FDB P value, and again replacing μ by the (first-level) bootstrap DGP, according to the bootstrap principle. The result is

$$p_3^f(\mu, \omega) \equiv R_0(Q^1(R^2(Q^1(p_2^f(\mu, \omega), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)), b(\mu, \omega)), \quad (30)$$

with p_2^f given by (24). The simulation estimate, which must be expressed as a function of the observed statistic t and bootstrap DGP β , is

$$\hat{p}_3^f(t, \beta) = \hat{R}_0(\hat{Q}^1(\hat{R}^2(\hat{Q}^1(\hat{p}_2^f(t, \beta), \beta), \beta), \beta), \beta), \quad (31)$$

with $\hat{p}_2^f(t, \beta)$ given by (25).

Here is the algorithm for the FTB P value.

Algorithm FTB:

1. From the data set under analysis, compute the realised statistic t and the bootstrap DGP β .
2. Draw B bootstrap samples and compute B bootstrap statistics $t_j^* = \tau(\beta, \omega_j^*)$, $j = 1, \dots, B$, and B iterated bootstrap DGPs $\beta_j^* = b(\beta, \omega_j^*)$.
3. Compute B second-level bootstrap statistics $t_j^{1*} = \tau(\beta_j^*, \omega_j^{**})$, and sort them in increasing order. At the same time, compute the corresponding second-level bootstrap DGPs $\beta_j^{**} = b(\beta_j^*, \omega_j^{**})$.
4. Compute B third-level bootstrap statistics $t_j^{2*} = \tau(\beta_j^{**}, \omega_j^{***})$.
5. Compute the estimated first-level bootstrap P value $\hat{p}_1(t, \beta)$, as the proportion of the t_j^* smaller than t .
6. Obtain the estimate $\hat{Q}^{1*} \equiv \hat{Q}^1(\hat{p}_1(t, \beta), \beta)$ as the order statistic of the t_j^{1*} of rank $\lceil B\hat{p}_1(t, \beta) \rceil$.
7. Compute the estimated FDB P value $\hat{p}_2^f(t, \beta)$ as the proportion of the t_j^* smaller than \hat{Q}^{1*} .
8. Compute $\hat{Q}^{1**} \equiv \hat{Q}^1(\hat{p}_2^f(t, \beta), \beta)$ as the order statistic of the t_j^{1*} of rank $\lceil B\hat{p}_2^f(t, \beta) \rceil$.
9. Compute $\hat{R}^{2*} \equiv \hat{R}^2(\hat{Q}^1(\hat{p}_2^f(t, \beta), \beta), \beta)$ as the proportion of the t_j^{2*} smaller than \hat{Q}^{1**} .
10. Compute $\hat{Q}^{1***} \equiv \hat{Q}^1(\hat{R}^2(\hat{Q}^1(\hat{p}_2^f(t, \beta), \beta), \beta), \beta)$ as the order statistic of the t_j^{2*} of rank $\lceil r\hat{R}^{2*} \rceil$.
11. Compute $\hat{p}_3^f(t, \beta)$ as the proportion of the t_j^* smaller than \hat{Q}^{1***} .

5. Fast Higher-Order Bootstraps

The ideas that lead to the FDB and FTB P values can obviously be extended to higher orders. For the FDB, we approximate the distribution of the first-level bootstrap P value $p_1(\mu, \omega)$, and evaluate it at the computed first-level P value $p_1(t, \beta)$ and the bootstrap DGP β . (To avoid notational clutter, we omit the hats from estimated quantities.) For the FTB, we approximate the distribution of the FDB P value $p_2^f(\mu, \omega)$ and evaluate it at the computed FDB P value $p_2^f(t, \beta)$ and β . For a fast quadruple bootstrap, we wish to approximate the distribution of the FTB P value $p_3^f(\mu, \omega)$ and evaluate it at the computed FTB P value $p_3^f(t, \beta)$ and β . And so on.

The approximate CDFs R_1^f and R_2^f are given explicitly by (23) and (29). We define higher-order approximate CDFs and fast higher-order bootstrap P values recursively, as follows:

$$R_k^f(\alpha, \mu) \approx \mathbb{E}[\mathbf{I}(p_k^f(\mu, \omega) < \alpha)], \text{ and} \quad (32)$$

$$p_{k+1}^f(\mu, \omega) = R_k^f(p_k^f(\mu, \omega), b(\mu, \omega)), \quad (33)$$

where the exact nature of the approximation in (32) above will now be made explicit.

For the following discussion, the notation t stands for $\tau(\mu, \omega)$, and β stands for $b(\mu, \omega)$. We need not distinguish between the random variables and their realisations. Thus $p_2^f(t, \beta)$ means $p_2^f(\mu, \omega)$, and so forth. By (23), the approximate CDF $R_1^f(\alpha, \mu)$ of $p_1(t, \beta)$ is $R_0(Q^1(\alpha, \mu), \mu)$. By (29), $R_2^f(\alpha, \mu)$ is $R_0(Q^1(R^2(Q^1(\alpha, \mu), \mu), \mu), \mu)$. As we will see, the pattern for R_k^f is the composition of 2^k functions with R and Q alternating. In order to see how to determine what these functions are, we consider explicitly the case of the fast quadruple bootstrap, which will let us describe the iterative procedure necessary for the explicit expression of R_k^f for general k .

The explicit expression of $p_3^f(t, \beta)$ is given by (30) and (31), and for present purposes it can be written as

$$p_3^f(t, \beta) = R_0(Q^1(R^2(Q^1(p_2^f(t, \beta), \beta), \beta), \beta), \beta). \quad (34)$$

In order to approximate its distribution, we define the random variable

$$\tau^3(\mu, \omega_1, \omega_2, \omega_3, \omega_4) = \tau(b(b(b(\mu, \omega_1), \omega_2), \omega_3), \omega_4).$$

Its CDF is readily seen to be

$$R^3(\alpha, \mu) = \mathbb{E}[R^2(\alpha, b(\mu, \omega))].$$

The corresponding quantile function, $Q^3(\alpha, \mu)$ is such that

$$\mathbb{E}[Q^2(\alpha, b(\mu, \omega))] = Q^3(\alpha, \mu)$$

with error small of second order. By extension, it is obvious how to define τ^k , R^k and Q^k . The CDF of $p_3^f(t, \beta)$ evaluated at α is $E[\mathbb{I}(p_3^f(t, \beta) < \alpha)]$. By use of (34), we see that the inequality in the indicator here can be expressed as

$$p_2^f(t, \beta) < R^1(Q^2(R^1(Q_0(\alpha, \beta), \beta), \beta), \beta). \quad (35)$$

The probability that this inequality is satisfied under μ , conditional on β , can be approximated by the approximate CDF R_2^f of $p_2^f(t, \beta)$ evaluated at the right-hand side of (35) and μ . By (29), this is

$$R_0(Q^1(R^2(Q^1(R^1(Q^2(R^1(Q_0(\alpha, \beta), \beta), \beta), \beta), \mu), \mu), \mu), \mu), \mu). \quad (36)$$

Then $R_3^f(\alpha, \mu)$ is the unconditional expectation of this expression.

An argument by now familiar shows that this unconditional expectation is

$$\begin{aligned} & E[R_0(Q^1(R^2(Q^1(R^1(Q^2(R^1(Q^1(\alpha, \mu), \beta), \beta), \beta), \mu), \mu), \mu), \mu))] = \dots \\ & = R_0(Q^1(R^2(Q^1((R^2(Q^3(R^2(Q^1(\alpha, \mu), \mu), \mu), \mu), \mu), \mu), \mu), \mu), \mu). \end{aligned} \quad (37)$$

If we compare the sequence of functions in the expression (37) of $R_3^f(\alpha, \mu)$ and the expression (29) of $R_2^f(\alpha, \mu)$, we see that there are exactly twice as many in the former as in the latter. This arises because the inequality (35) has the four functions of p_2^f , and they are then the final four in (36), preceded by the four functions of R_2^f . We may observe that in (36) the composition of the final four functions is the inverse of the composition of the first four. When we get to (37), the indices of the final four functions have all been raised by 1.

By the definition (33), the fourth-level P value $p_4^f(t, \beta)$ is $R_3^f(p_3^f(t, \beta), \beta)$. It is not hard to check that, in the explicit expression of p_4^f , we have, first, the eight functions in (37), followed by the inverse of their composition with indices raised by 1, that is, the composition of the sixteen functions that we write in sequence as follows:

$$R_0 Q^1 R^2 Q^1 R^2 Q^3 R^2 Q^1 R^2 Q^3 R^4 Q^3 R^2 Q^3 R^2 Q^1.$$

Although the way in which we have arrived at this sequence of functions is easy enough to describe, the explicit structure seems not to be expressible in closed form other than by actually working it out.

The following algorithm provides an equivalent but slightly easier way to derive the sequence of functions in R_{k+1}^f or p_{k+1}^f from the sequence in R_k^f and p_k^f .

Algorithm FkB:

1. Divide the sequence S_k of the 2^k functions in R_k^f into two sequences A_k and B_k , of length 2^{k-1} each, such that $S_k = A_k B_k$.
2. Obtain the sequence C_k of functions the composition of which is the inverse of the composition of the functions in the sequence B_k .
3. Obtain a new sequence D_k by incrementing the indices of the elements of the sequence C_k by 1.
4. The sequence S_{k+1} used to define R_{k+1}^f and p_{k+1}^f is $A_k B_k D_k B_k$, of length 2^{k+1} .

6. Illustrations

In this section, we present the results of various simulation experiments designed to see to what extent the fast double and triple bootstraps can improve the reliability of inference. The first experiments, which deal with a test for a unit root, make use of a parametric bootstrap that makes the distributions of bootstrap statistics absolutely continuous. The other two sets of experiments, one dealing with a test for an ARCH effect, the other a test for serial correlation of the disturbances in a regression model, use resampling bootstraps, which lead to bootstrap statistics with discrete distributions, in violation of one of the assumptions made in the earlier part of the paper. Since resampling is far and away the most frequently used form of bootstrapping in empirical work, it is highly desirable to see if the fast iterated bootstraps suffer noticeably from the discrete distributions induced by resampling.

Testing for a unit root

There are well-known difficulties in testing for a unit root in a series obtained by summing a stationary series that is an MA(1) process with a parameter θ close to -1. Unless special precautions are taken, augmented Dickey-Fuller (ADF) tests can suffer from severe overrejection under the null hypothesis of a unit root, on account of the near cancellation of the unit root by the MA component in the driving stationary series. We may cite Schwert (1989) and Perron and Ng (1996) in this regard.

Over the last decade or so, various bootstrap techniques have been proposed as a means of improving size distortions in the unit-root testing literature. One bootstrap technique designed to deal with autoregressive models is the so called sieve bootstrap first proposed by Bühlmann (1997). Simulation evidence demonstrates that this bootstrap approach has certain appeal in reducing size distortions. Here it is appropriate to cite Psaradakis (2001), Chang and Park (2002), and Palm, Smeekes, and Urbain (2008) as evidence of this fact. These papers show the ability of bootstrapped ADF statistics to outperform their asymptotic counterparts. However, despite the ability of the sieve bootstrap to reduce size distortions in certain cases, the gain is really unimpressive in the case considered in our simulations, in which the MA component has a parameter close to -1. In this regard, Richard (2009) applies several variations of the FDB to show that size distortions can be significantly reduced by imposing certain linear restrictions on the truncation parameter of the bootstrap ADF regression and the truncation parameter of the bootstrap sieve.

In Davidson (2010), it is shown that, under the assumption that the MA(1) process has Gaussian innovations, parametric FDB tests can significantly reduce the size distortion of an ADF test. It is argued that bootstrapping the FDB should reduce the distortions still further. This is of course very computationally demanding. In what follows, we show that comparable results may be obtained far more economically with the fast triple bootstrap.

The model studied in this section may be summarised as follows:

$$y_t = \rho y_{t-1} + v_t \tag{38}$$

$$v_t = u_t + \theta u_{t-1}, \quad u_t \sim \text{NID}(0, \sigma^2), \quad t = 1, \dots, n. \tag{39}$$

The observed series is y_t , and the null hypothesis of a unit root sets $\rho = 1$. Under that hypothesis, $v_t = \Delta y_t$, where Δ is the first-difference operator. We may write (39) in vector notation using the lag operator L , as follows:

$$\mathbf{v} = (1 + \theta L)\mathbf{u}, \quad \text{or} \quad \mathbf{v} = R(L)\mathbf{v} + \mathbf{u},$$

where we define $R(L) = \theta(1 + \theta L)^{-1}L$. Davidson (2010) demonstrates that θ may be estimated by a two-stage nonlinear least squares regression using the model:

$$\mathbf{v} = \phi \mathbf{e}_1 + \theta(1 + \theta L)^{-1}L(\mathbf{v} - \phi \mathbf{e}_1) + \mathbf{u}, \quad (40)$$

where \mathbf{e}_1 is the vector with first element equal to 1 and all others zero. In the first stage, ϕ is set to zero and a preliminary estimate of θ is obtained. Then an estimate of ϕ is given by $\mathbf{s}^\top(1 - R(L))\mathbf{u}/\mathbf{s}^\top\mathbf{s}$, where $\mathbf{s} = (1 - R(L))\mathbf{e}_1$. Finally, this estimate of ϕ is used as a known constant in (40), which is re-estimated to obtain the second-stage estimate of θ .

Testing for a unit root in (38) proceeds by computing an ADF statistic using the ADF regression:

$$\Delta y_t = \beta_0 + \beta_1 y_{t-1} + \sum_{i=1}^p \gamma_i \Delta y_{t-i} + \text{residual}. \quad (41)$$

The statistic is the standard t -statistic for the null hypothesis $\beta_1 = 0$ when (41) is estimated by ordinary least squares. The ADF statistic, although easy to compute, has a non-standard asymptotic distribution which is that of a functional of Brownian motion that depends on no nuisance parameters. The ADF statistic is thus an asymptotic pivot.

A suitable bootstrap DGP can be expressed as follows:

$$\mathbf{u}^* \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{v}^* = (1 + \hat{\theta}L)\mathbf{u}^*, \quad \mathbf{y}^* = (1 - L)^{-1}\mathbf{v}^*, \quad (42)$$

where $\hat{\theta}$ is obtained by the procedure outlined above, with $\mathbf{v} = \Delta \mathbf{y} = (1 - L)\mathbf{y}$. Note that, since the ADF statistic is scale-invariant, we can set $\sigma = 1$. Thus the bootstrap DGP is completely determined by one single parameter, the estimate $\hat{\theta}$.

In the simulation experiments we now discuss, the data were generated by the model defined by (38) and (39), with $\rho = 1$ and for various values of θ close to -1. For any given estimate $\hat{\theta}$, the bootstrap DGP (42) was used. Second-level bootstrap DGPs were formulated by generating bootstrap data with $\hat{\theta}$ in (39), and then obtaining the estimate of θ for these data. This estimate, which we denote $\hat{\theta}^*$, then characterises a second-level bootstrap DGP, which can subsequently be used to generate second-level bootstrap data, used to obtain an estimate $\hat{\theta}^{**}$, which characterises a third-level bootstrap DGP. This experimental design is the same as the one used in Davidson (2010).

Figures 1 and 2 below show the errors in rejection probability (ERP), that is, the difference between the experimentally observed rejection rate and the nominal level α , of the ordinary single bootstrap, the FDB, and the FTB for varying values of the MA(1) parameter θ , for sample sizes $n = 50$ and 100 , and for all nominal levels $0 \leq \alpha \leq 1$. Although for practical

purposes one need not consider values of α greater than 0.1, using all possible values allows a clearer view of how the distributions of the various bootstrap P values differ from the ideal $U(0,1)$ distribution. All graphs in Figure 1 were computed with $N = 25,000$ replications, $B = 4,999$ bootstrap repetitions, and ADF truncation parameter $p = 12$. The time required to complete the simulations using a computer cluster with 54 nodes with 4 CPUs each was roughly 16 hours and 22 hours for sample sizes $n = 50$ and $n = 100$ respectively.

We see that the FTB P values tend to have lower ERPs than their FDB and single bootstrap (BS) counterparts. As expected, the size correction is not very impressive for $\theta = -0.99$, but as θ increases away from $\theta = -1$ improvements can be drastic. What is interesting however is that when θ is very close to -1 as in Figure 1, the FTB ERP is slightly worse than the FDB ERP, which in turn is slightly worse than the single bootstrap ERP in the region of the nominal size which is most relevant to practitioners, namely $0 \leq \alpha \leq 0.10$.

An unexpected and somewhat strange feature of the results shown in the figures is that the distortions for $n = 50$ are often smaller than those for $n = 100$. This is just another manifestation of the failure of inference in the close vicinity of $\theta = -1$.

Figures 3 and 4 are like Figures 1 and 2, but they omit the case of the single bootstrap, and include results for the standard double and triple bootstraps. They show that the behaviour of the fast iterated bootstraps is very similar to that of the standard iterated bootstraps. The latter are slightly less distorted than the former, but the fast triple bootstrap usually outperforms the standard double bootstrap. Some caution is required in comparing the experimental results in Figures 3 and 4, because, for reasons of computational feasibility, the results for the standard iterated bootstraps were obtained with $N = 12,500$, $B = 99$, $B_1 = 119$, and $B_2 = 129$. in order to complete the simulations in a reasonable time frame. Even so, using all 216 CPUs of the cluster, it took over 24 hours to complete a simulation for a single value of θ with a sample size $n = 50$. But, even allowing for non-negligible simulation randomness, the ordering of the various procedures is clearly established.

Before ending this section, we must remark that the model considered here is much too restrictive to have any practical interest. We have studied it as a computationally feasible test bed for the fast and standard iterated bootstraps, and to demonstrate that going as far as the fast triple bootstrap does indeed have the possibility of giving improved reliability over the single and fast double bootstraps.

A test for ARCH

In Davidson and MacKinnon (2007), one of the examples used to show how the FDB improves on the single bootstrap is a test for ARCH disturbances in a linear regression model. Since the seminal work of Engle (1982), it has been recognized that serial dependence in the variance of the disturbances of regression models using time-series data is a very common phenomenon. It is therefore usually advisable to test for the presence of such serial dependence.

Consider the linear regression model

$$\begin{aligned} y_t &= \mathbf{X}_t \boldsymbol{\beta} + u_t, & u_t &= \sigma_t \varepsilon_t, & t &= 1, \dots, n, \\ \sigma_t^2 &= \sigma^2 + \gamma u_{t-1}^2 + \delta \sigma_{t-1}^2, & \varepsilon_t &\sim \text{IID}(0, 1). \end{aligned} \tag{43}$$

The disturbances of this model follow the GARCH(1,1) process introduced by Bollerslev (1986). The easiest way to test the null hypothesis that the u_t are IID in the model (43) is to run the regression

$$\hat{u}_t^2 = b_0 + b_1 \hat{u}_{t-1}^2 + \text{residual}, \tag{44}$$

where \hat{u}_t is the t^{th} residual from an OLS regression of y_t on \mathbf{X}_t . The null hypothesis that $\gamma = \delta = 0$ can be tested by testing the hypothesis that $b_1 = 0$. Besides the ordinary t statistic for b_1 , a commonly used statistic is n times the centred R^2 of the regression, which has a limiting asymptotic distribution of χ_1^2 under the null hypothesis.

Since in general one is unwilling to make any restrictive assumptions about the distribution of the ε_t , a resampling bootstrap seems the best choice. This is in violation of one of the main assumptions in this paper, namely that the distribution of the statistic is absolutely continuous with respect to Lebesgue measure on the real line. Resampling gives rise to a discrete distribution, although, for samples of reasonable size, it may be close enough to being continuous for the discreteness not to matter. It is of course of interest to see to what extent the theory of fast iterated bootstraps can be used effectively with resampling. Alternatively, the discrete distribution of the resampled objects can be smoothed. Davidson and MacKinnon tried this with the model (43), and found that it made a noticeable, but very slight, difference to the performance of both single and fast double bootstraps. In the experiments described in this section, we do not smooth.

The experimental design is copied from Davidson and MacKinnon (2007). In all cases, \mathbf{X}_t consists of a constant and two independent, standard normal random variates, since changing the number of regressors has only a modest effect on the finite-sample behaviour of the tests. The sample size takes on the values 40, 80, and 160, a small subset of the set of sample sizes studied by Davidson and MacKinnon. In order to have non-negligible ERPs, the ε_t are drawn from the χ_2^2 distribution, subsequently centred and rescaled to have variance 1. Without loss of generality, we set $\boldsymbol{\beta} = \mathbf{0}$ and $\sigma^2 = 1$, since the test statistic is invariant to changes in the values of these parameters.

The invariance means that we can use as bootstrap DGP the following:

$$y_t^* = u_t^*, \quad u_t^* \sim \text{EDF}(y_t),$$

where the notation EDF (for “empirical distribution function”) means simply that the bootstrap data are resampled from the original data. For iterated bootstraps, y_t^{**} is resampled from the y_t^* , and y_t^{***} is resampled from the y_t^{**} .

The experiments consisted of $N = 10,000$ replications with $B = 399$ bootstrap repetitions each. For each replication, P values were computed for the asymptotic test, the single bootstrap, the FDB, and the FTB. The results under the null are displayed in Figure 5.

Observe that there is little difference in the ERPs for $n = 80$ and $n = 160$. The ordering of the four procedures is quite clear. The single bootstrap improves considerably on the asymptotic test based on the χ_1^2 distribution, the FDB again considerably on the single bootstrap, and the FTB has no significant distortion at all for $n = 80$ and $n = 160$, and very little for $n = 40$.

One might wonder if the excellent behaviour of the FTB is bought at the cost of diminished power. That this is not the case is demonstrated by the experiments the results of which are shown in [Figure 6](#). The simulated data were generated by the DGP of model (43) with $\sigma^2 = 1$, $\gamma = 0.3$, and $\delta = 0.3$. These values were chosen in order to have some power for $n = 40$ and not to have so much power for $n = 160$ for an ordering of the results to be unclear. In this figure, the experimental rejection rate is plotted as a function of the *nominal* level of the test; see Horowitz and Savin (2000) and Davidson and MacKinnon (2006) for discussion of why this makes more sense than attempting any sort of “size adjustment”.

The under-rejection of the asymptotic test under the null is carried over under the DGP studied here, to the extent that, for $n = 40$, the rejection rate of the test is smaller than the nominal level for some values of the latter. This is no longer true for the larger values of n . The three bootstrap tests have very similar rejection rates, always greater than that of the asymptotic test, and with the FDB and FTB having slightly greater power than the single bootstrap test.

A test for serial correlation

Another of the examples of the good performance of the FDB found in Davidson and MacKinnon (2007) is given by the Durbin-Godfrey test for serial correlation of the disturbances in a linear regression model. The test was introduced in Durbin (1970) and Godfrey (1978). The model that serves as the alternative hypothesis for the test is the linear regression model

$$y_t = \mathbf{X}_t\boldsymbol{\beta} + \gamma y_{t-1} + u_t, \quad u_t = \rho u_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{IID}(0, \sigma^2), \quad t = 1, \dots, n, \quad (45)$$

where \mathbf{X}_t is a $1 \times k$ vector of observations on exogenous variables. The null hypothesis is that $\rho = 0$. Let the OLS residuals from running regression (45) be denoted \hat{u}_t . Then the Durbin-Godfrey (DG) test statistic is the t statistic for \hat{u}_{t-1} in a regression of y_t on \mathbf{X}_t , y_{t-1} , and \hat{u}_{t-1} . It is asymptotically distributed as $N(0, 1)$ under the null hypothesis. Since this test can either overreject or underreject in finite samples, it is natural to use the bootstrap in an effort to improve its finite-sample properties.

For the bootstrap DGP, from running regression (45), we obtain estimates $\hat{\boldsymbol{\beta}}$, $\hat{\gamma}$, as well as the residuals \hat{u}_t . The semiparametric bootstrap DGP can be written as

$$y_t^* = \mathbf{X}_t\hat{\boldsymbol{\beta}} + \hat{\gamma}y_{t-1}^* + u_t^*, \quad (46)$$

where the u_t^* are obtained by resampling the residuals rescaled as $(n/(n - k - 1))^{1/2}\hat{u}_t$. The initial value y_0^* is set equal to the actual pre-sample value y_0 .

Our experimental design is similar to that in Davidson and MacKinnon (2007). We set $k = 6$, with the first regressor a constant, and the remaining five generated by independent, stationary AR(1) processes with normal innovations and parameter $\rho_x = -0.8$. The disturbances ε_t are normally distributed with $\sigma = 10$. We put $\beta = \mathbf{0}$ and $\gamma = 0.75$. We look at sample sizes $n = 20$ and 40 under the null, and also at $n = 56$ in our study of power. These choices are sufficient for us to distinguish clearly the behaviour of the various testing procedures: asymptotic, single bootstrap, FDB, and FTB. As before, we used 10,000 replications each involving 399 bootstrap repetitions.

Figure 7 shows the ERPs under the null. For $n = 40$, we do not show the ERP of the asymptotic test, because it is so great that including it in the graph would make the ERPs of the other tests indistinguishable. All the bootstrap procedures are much less distorted than the asymptotic test, and they have, once again, the same ordering as before, with the FTB the least distorted. Even for as small a sample size as 40, its observed rejection rate is never different from the nominal level by more than 0.01.

Power is illustrated in Figure 8. The data for these experiments were generated with the autocorrelation parameter ρ in (45) equal to 0.5, $\sigma = 1$, and the parameter ρ_x used in generating the regressors equal to 0.8. These values were chosen on the basis of the results in Davidson and MacKinnon (2007), where they give rise to significant differences in the power of the single bootstrap and the FDB. For $n = 20$, there is no visible power at all; in fact there is some slight underrejection by all the tests. With $n = 40$, the asymptotic test rejects slightly more often than the bootstrap tests, and, for $n = 56$, all tests have very similar rejection rates. Once again, the superior performance of the fast iterated bootstrap tests does not entail reduced power.

7. Conclusions

The approximations that led to the fast double bootstrap have been extended not only to a fast triple bootstrap but to arbitrary levels of bootstrap iteration. Algorithms have been given for the implementation of the fast double and fast triple bootstraps, along with an algorithm for generating the expressions to be estimated by simulation for higher-level iterated bootstraps.

The approximations underlying the fast iterated bootstraps are justified only under two assumptions. The first is that the statistic being bootstrapped is asymptotically independent of the bootstrap DGP, and the second is that all bootstrap DGPs generate statistics of which the distributions are absolutely continuous with respect to Lebesgue measure on the real line. Although the second assumption is not satisfied if a resampling bootstrap is used, the fast triple bootstrap is shown to suffer from less size distortion under the null than either the standard or the fast double bootstrap in a couple of simulation experiments. In the experiments with a unit root test, a parametric bootstrap is used, so that the second assumption is satisfied, but the first decidedly is not. Even so, the experiments show that the fast double and triple bootstraps suffer from only a little more distortion than their standard counterparts, with the same ranking of the various procedures as seen

in the resampling examples. Further, the resampling examples demonstrate that improved behaviour under the null is not achieved at the cost of reduced power.

The few experiments we report about the standard triple bootstrap required enormous computing resources. With current technology, and indeed with technology likely to be available in the near future, experiments on standard iterated bootstraps beyond the triple would consume computing power well beyond that accessible to academic researchers. The fast variants are much less computationally intensive, and their demands increase only linearly with the level of iteration, while those of the standard iterated bootstraps increase exponentially. For a single bootstrap P value, one has to evaluate $1 + B$ statistics and just one bootstrap DGP. For the level- k fast iterated bootstrap, one needs $1 + kB$ statistics, and $1 + (k - 1)B$ DGPs. But for the level- k standard iterated bootstrap, if for simplicity we do not use different numbers of bootstraps at each level, the number of statistics needed is $(B^{k+1} - 1)/(B - 1)$ and the number of DGPs is $(B^k - 1)/(B - 1)$.

Interesting theoretical questions remain, to do with the convergence or otherwise of the sequence of iterated bootstrap P values, fast and standard. It would be good to have conditions that guarantee convergence or non-convergence. Where convergence does occur, is the limiting distribution the uniform distribution on $[0, 1]$? With resampling, we know that it cannot be, since, with repeated resampling, eventually a stage is reached in which only one element is resampled. We hope to clarify these and other points in future work.

References

- Beran, R. (1987). “Prepivoting to reduce level error of confidence sets”, *Biometrika*, **74**, 457-68.
- Beran, R., (1988). “Prepivoting test statistics: a bootstrap view of asymptotic refinements”, *Journal of the American Statistical Association*, **83**, 687-697.
- Bollerslev, T. (1986). “Generalized autoregressive conditional heteroskedasticity,” *Journal of Econometrics*, **31**, 307–27.
- Bühlmann, P., (1997). “Sieve bootstrap for time series”, *Bernoulli*, **3**, 123-48.
- Chang, Y. and Park, J. Y., (2002). “A sieve bootstrap for the test of a unit root”, *Journal of Time Series Analysis* **24**, 379-400.
- Davidson, J. (2006). “Alternative bootstrap procedures for testing cointegration in fractionally integrated processes,” *Journal of Econometrics*, **133**, 741-777.
- Davidson, R. (2010). “Size Distortion of Bootstrap Tests: Application to a Unit Root Test”, *Review of Economic Analysis*, **2**, 169–193.
- Davidson, R. and J. G. MacKinnon (2002). “Fast Double Bootstrap Tests of Nonnested Linear Regression Models”, *Econometric Reviews*, **21**, 419-429.

- Davidson, R. and J. G. MacKinnon (2006). “The Power of Asymptotic and Bootstrap Tests”, *Journal of Econometrics*, **133**, 421–441.
- Davidson, R. and J. G. MacKinnon (2007). “Improving the Reliability of Bootstrap Tests with the Fast Double Bootstrap”, *Computational Statistics and Data Analysis*, **51**, 3259–3281.
- Durbin, J. (1970). “Testing for serial correlation in least-squares regression when some of the regressors are lagged dependent variables,” *Econometrica*, **38**, 410–421.
- Efron, B. (1979). “Bootstrap methods: Another look at the jackknife”, *Annals of Statistics*, **7**, 1–26.
- Engle, R. F. (1982). “Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica*, **50**, 987–1007.
- Godfrey, L. G. (1978). “Testing against general autoregressive and moving average error models when the regressors include lagged dependent variables,” *Econometrica*, **46**, 1293–1301.
- Hall, P. (1986). “On the bootstrap and confidence intervals”, *Annals of Statistics*, **14**, 1431–52.
- Hall, P. and M. A. Martin (1988). “On bootstrap resampling and iteration”, *Biometrika*, **75**, 661–71.
- Ho, Y. H. S. and S. M. S. Lee (2005). “Iterated smoothed bootstrap confidence intervals for population quantiles”, *Annals of Statistics*, **33**, 437–62.
- Horowitz, J. L. and N. E. Savin (2000). “Empirically relevant critical values for hypothesis tests”, *Journal of Econometrics*, **95**, 375–389.
- Lamarche, J.-F. (2004). “The numerical performance of fast bootstrap procedures,” *Computational Economics*, **23**, 379–389.
- Lee, S. M. S. and G. A. Young (1995). “Asymptotic iterated bootstrap confidence intervals”, *Annals of Statistics*, **23**, 1301–30.
- Lee, S. M. S. and G. A. Young (2003). “Prepivoting by weighted bootstrap iteration”, *Biometrika*, **90**, 393–410.
- Omtzigt, P., and S. Fachin (2002). “Bootstrapping and Bartlett corrections in the cointegrated VAR model”, University of Amsterdam Discussion Paper No. 2002/15.
- Ouyse, R. (2009). “Fast iterated bootstrap mean bias correction”, working paper in the Proceedings of 2009 NZESG.
- Palm, F. C., S. Smeekes, and J.-P. Urbain (2008). “Bootstrap unit root tests: comparison and extensions”, *Journal of Time Series Analysis*, **29**, 371–401.

- Perron, P. and S. Ng (1996). “Useful modifications to unit root tests with dependent errors and their local asymptotic properties”, *Review of Economic Studies* **63**, 435–65.
- Psaradakis, Z. (2001). “Bootstrap Tests for an Autoregressive Unit Root in the Presence of Weakly Dependent Errors”, *Journal of Time Series Analysis* **22**, 577-594.
- Richard, P. (2009). “Modified Fast Double Sieve Bootstraps for ADF Tests”, *Computational Statistics & Data Analysis*, **53**, 4490-4499.
- Schwert, G. W. (1989). “Testing for unit roots: a Monte Carlo investigation”, *Journal of Business and Economic Statistics* **7**, 147–59.

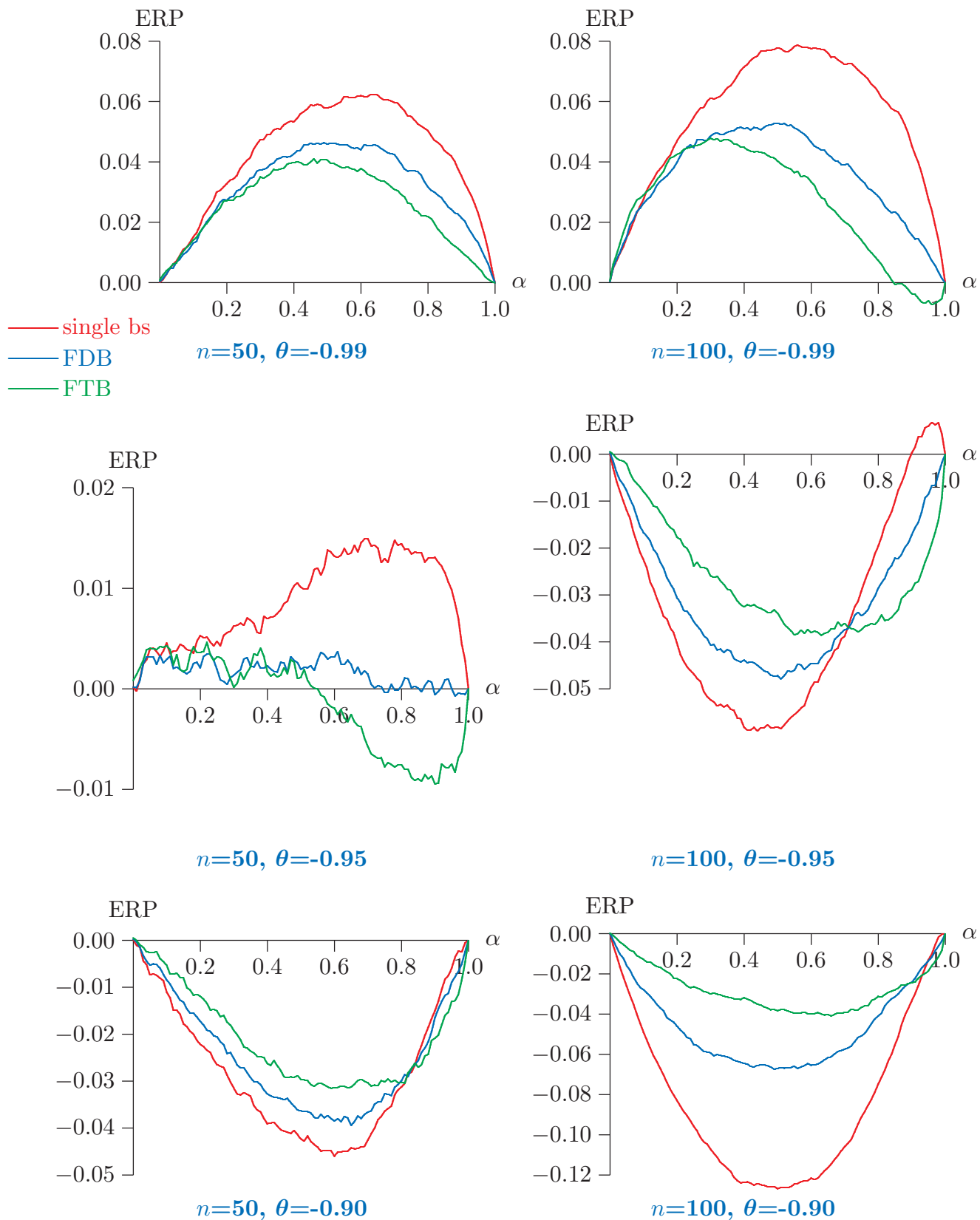
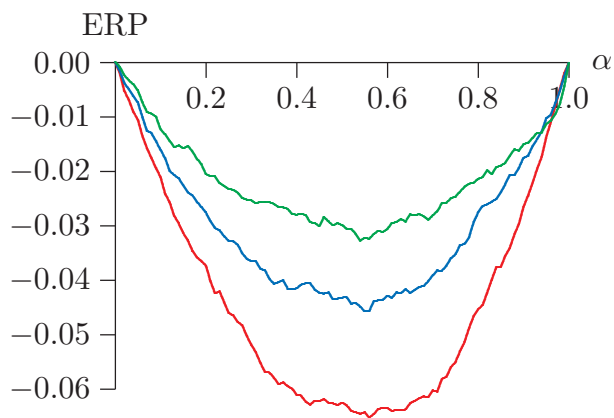
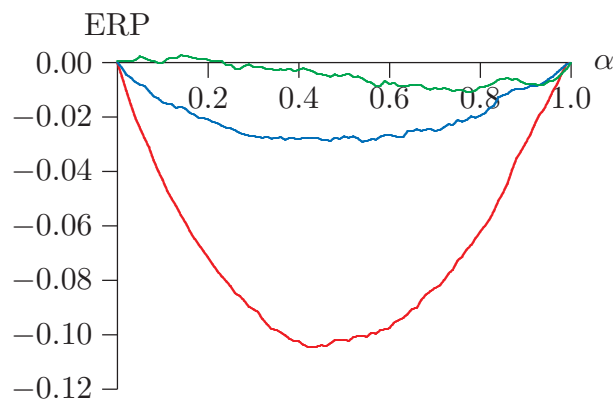


Figure 1: ERPs of unit root test



$n=50, \theta=-0.85$

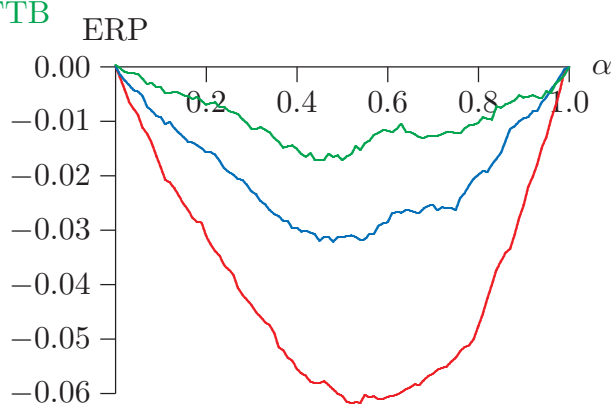


$n=100, \theta=-0.85$

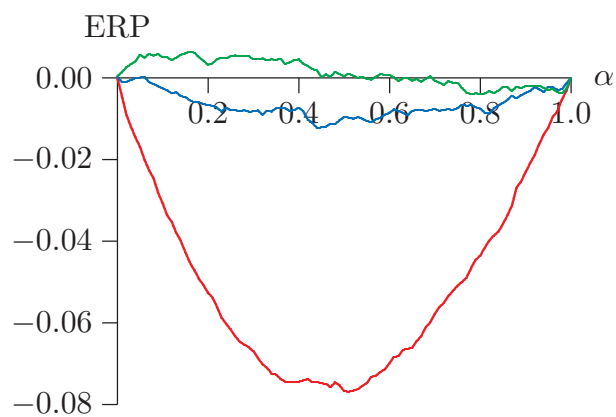
— single bs

— FDB

— FTB

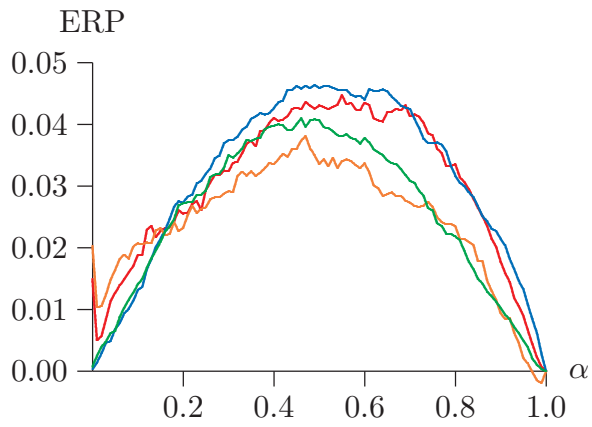


$n=50, \theta=-0.80$

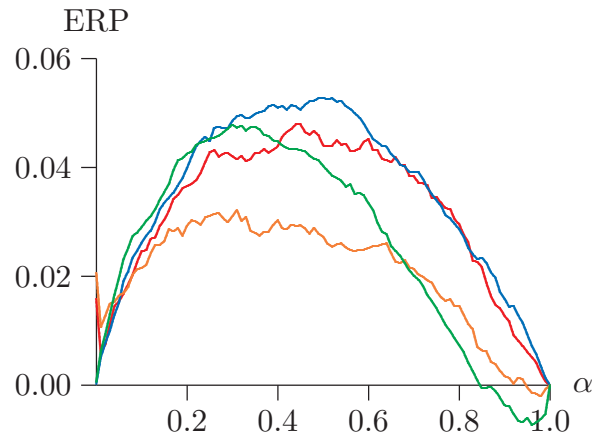


$n=100, \theta=-0.80$

Figure 2: ERPs of unit root test

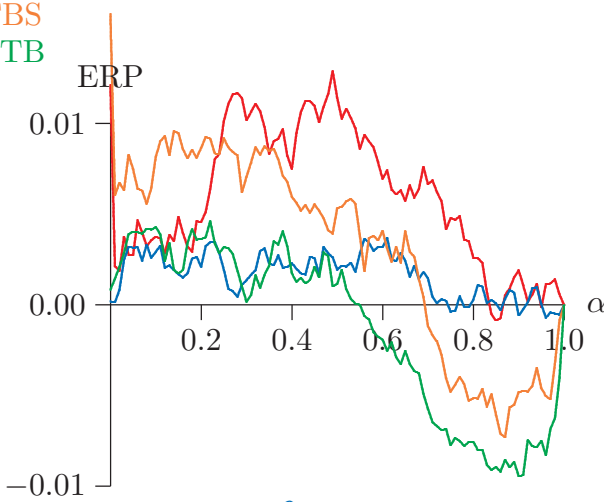


$n=50, \theta=-0.99$

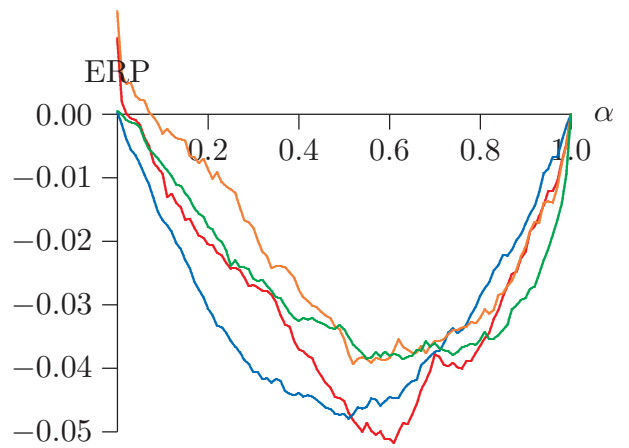


$n=100, \theta=-0.99$

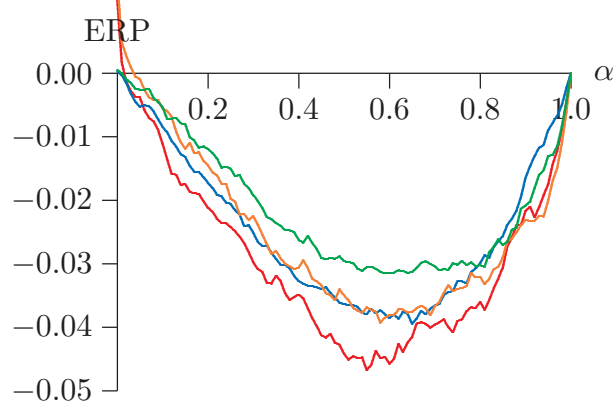
- DBS
- FDB
- TBS
- FTB



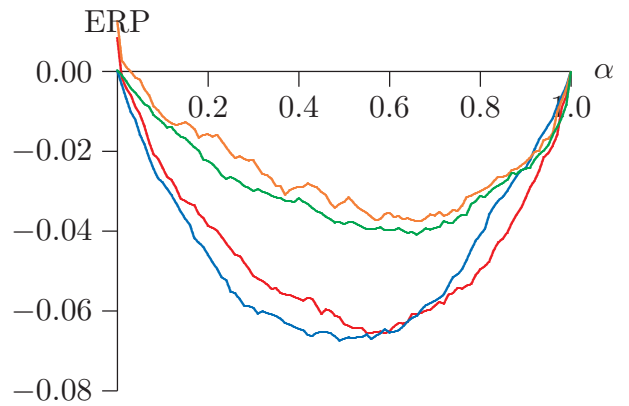
$n=50, \theta=-0.95$



$n=100, \theta=-0.95$

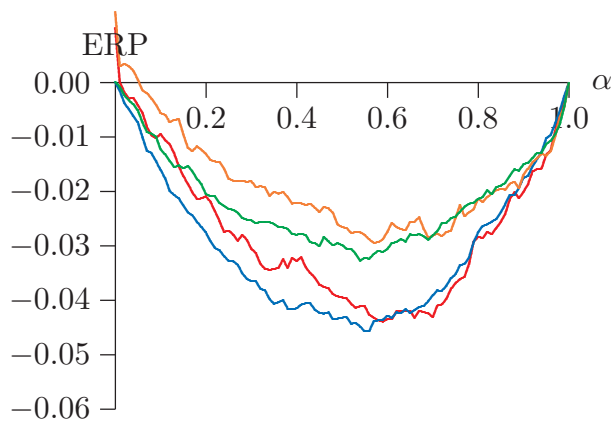


$n=50, \theta=-0.90$

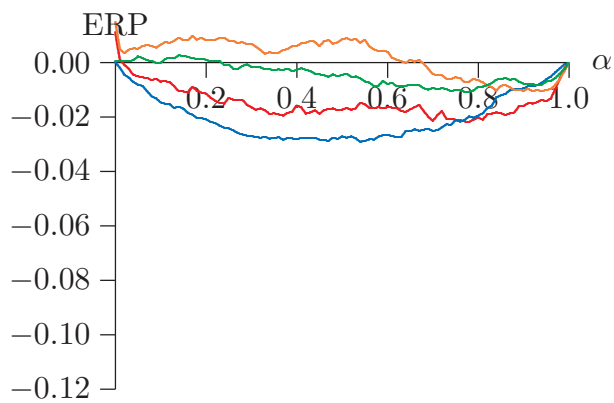


$n=100, \theta=-0.90$

Figure 3: ERPs of unit root test

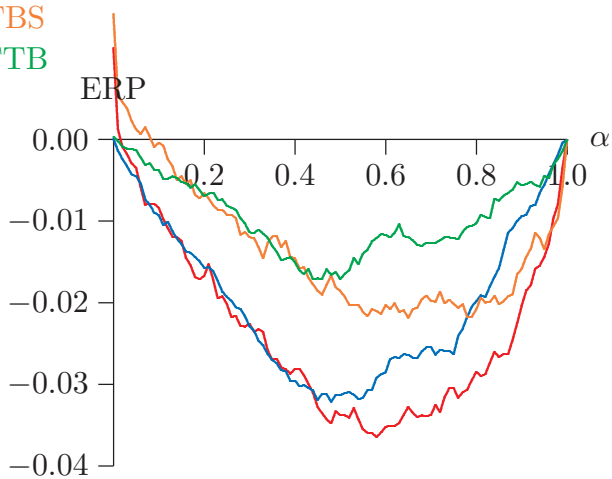


$n=50, \theta=-0.85$

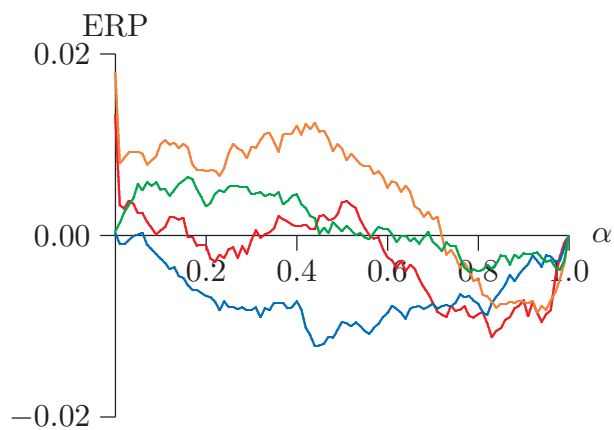


$n=100, \theta=-0.85$

— DBS
— FDB
— TBS
— FTB



$n=50, \theta=-0.80$



$n=100, \theta=-0.80$

Figure 4: ERPs of unit root test

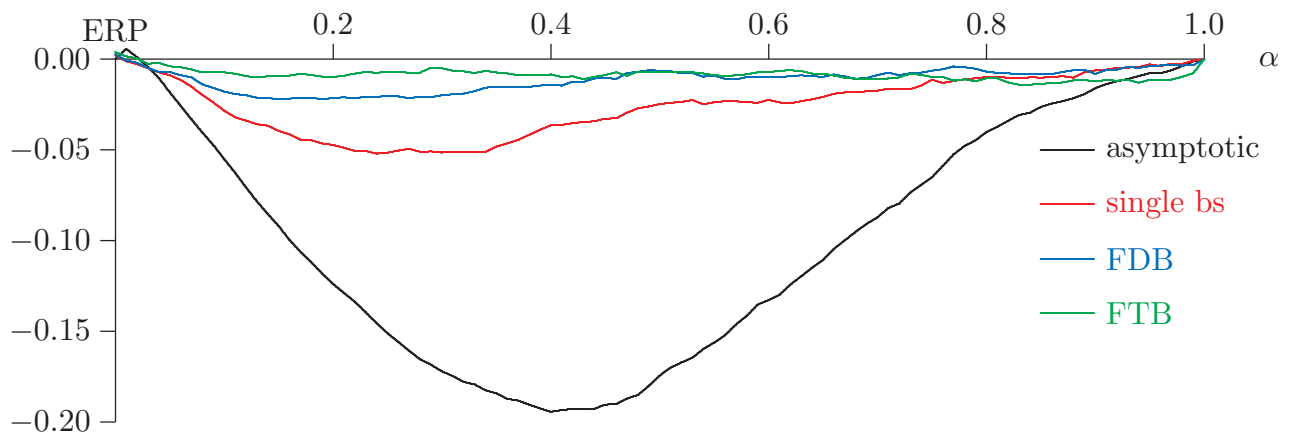
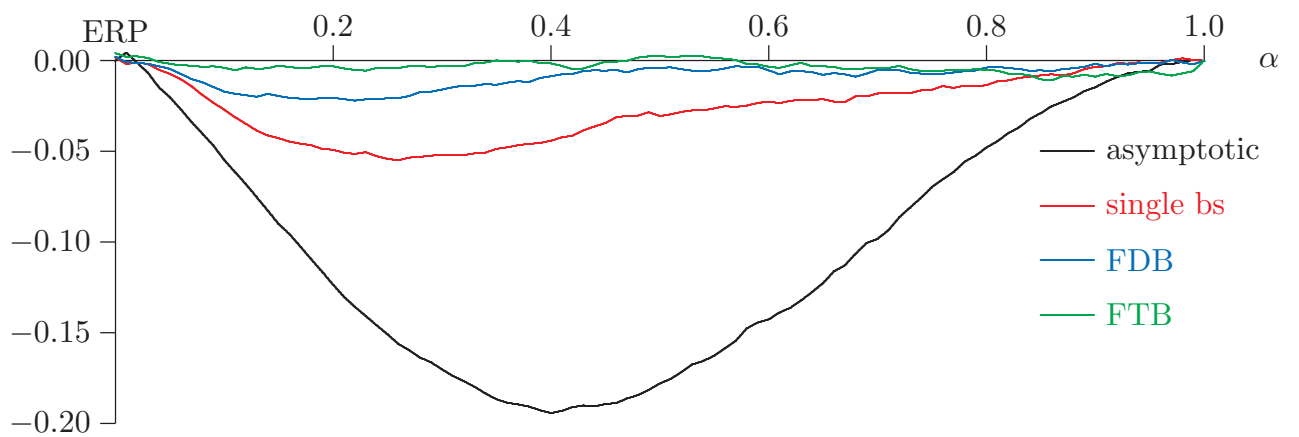
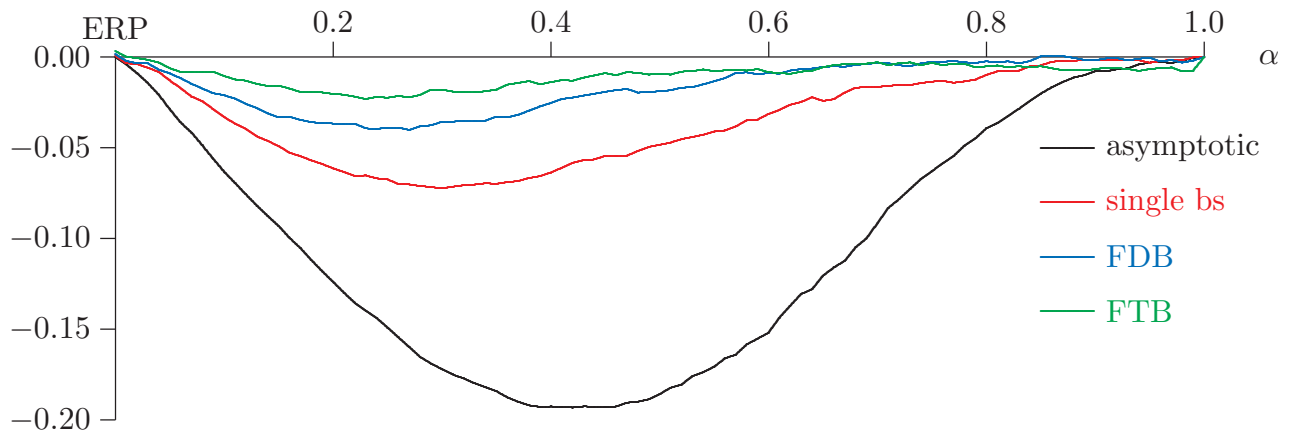
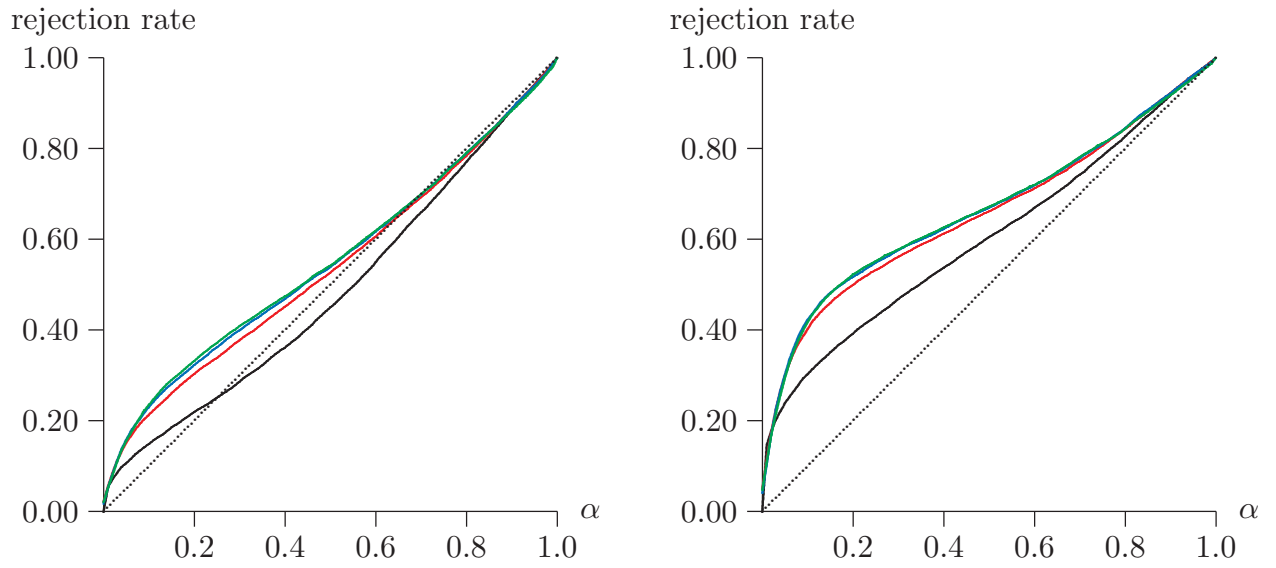
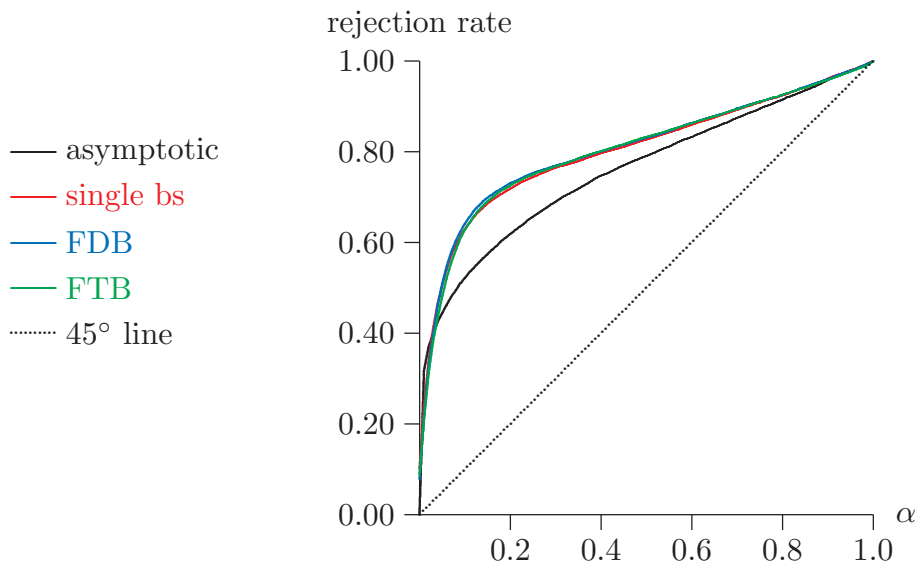


Figure 5: ERPs of test for ARCH



$n=40$

$n=80$



$n=160$

Figure 6: ARCH test rejection rate, $\alpha=1$, $\gamma=\delta=0.3$

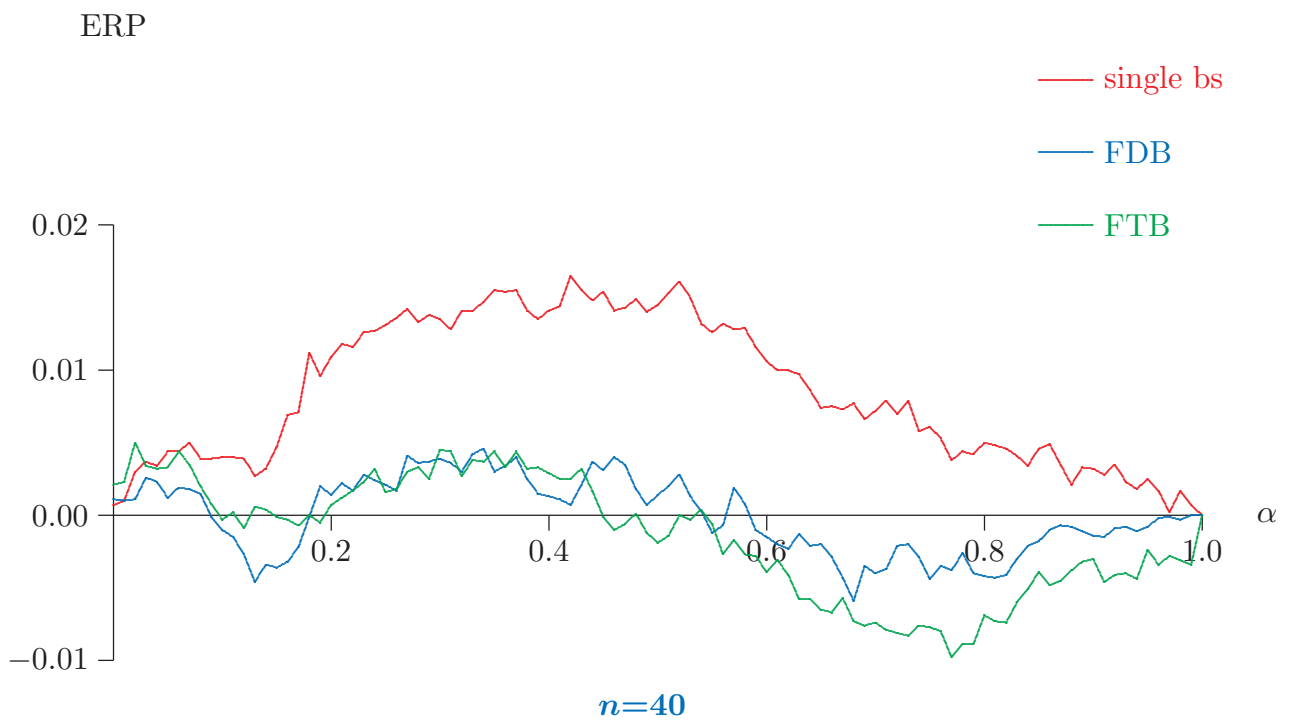
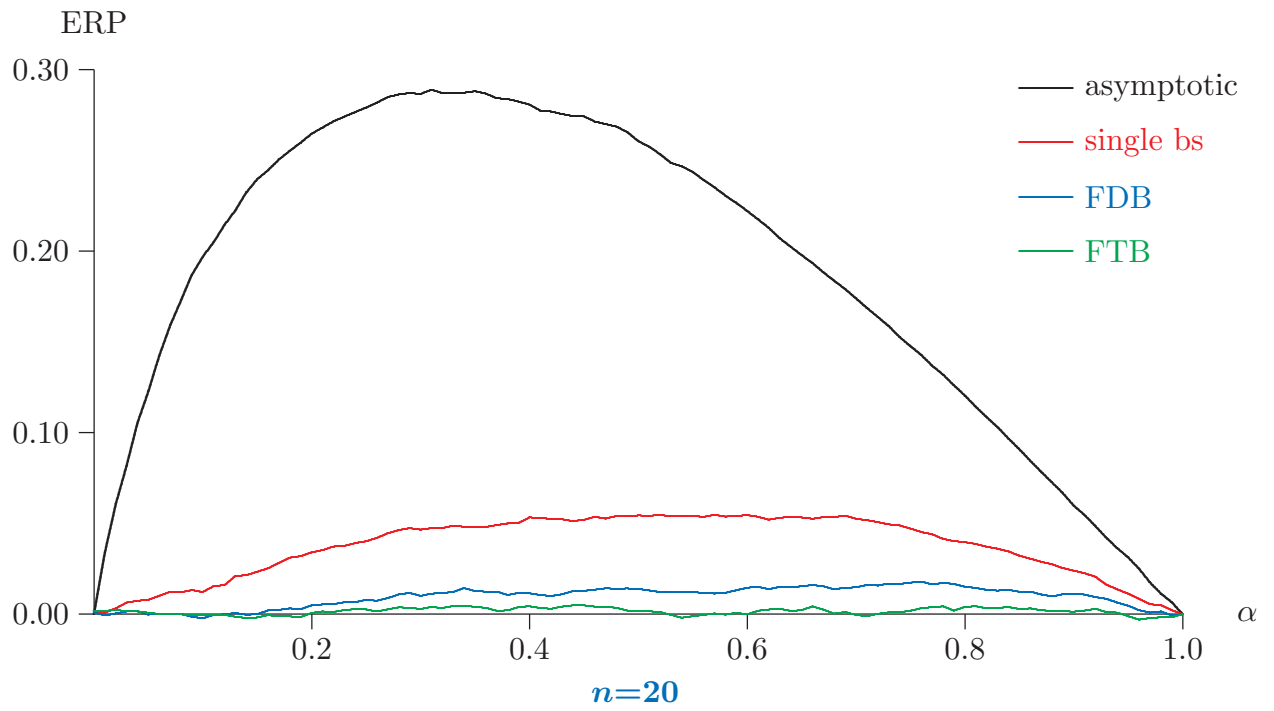
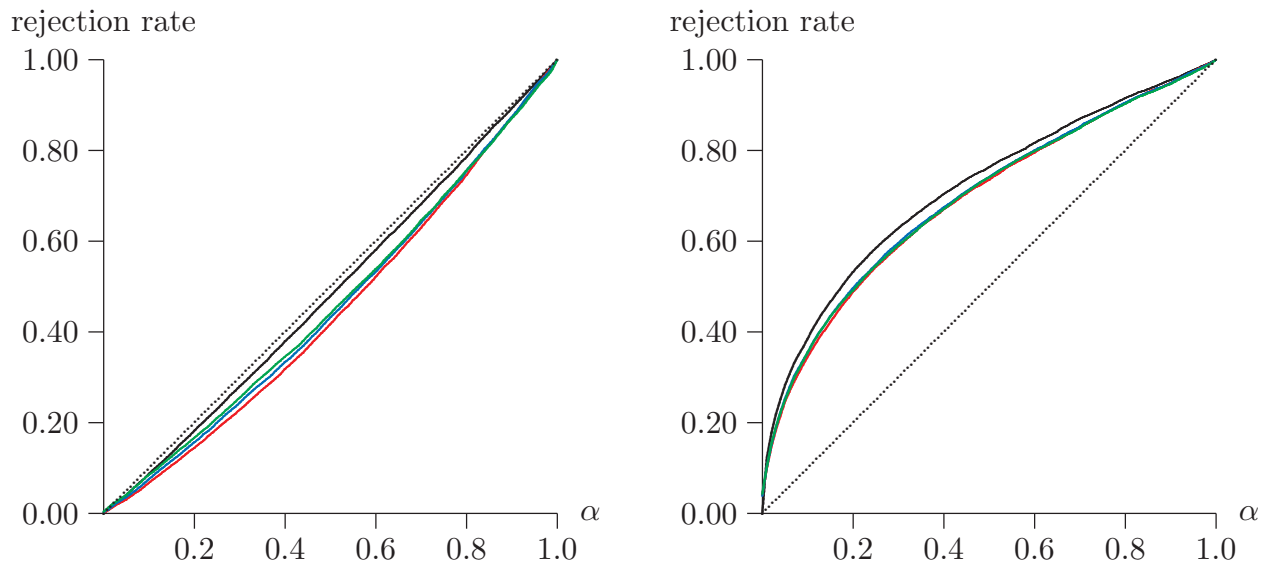
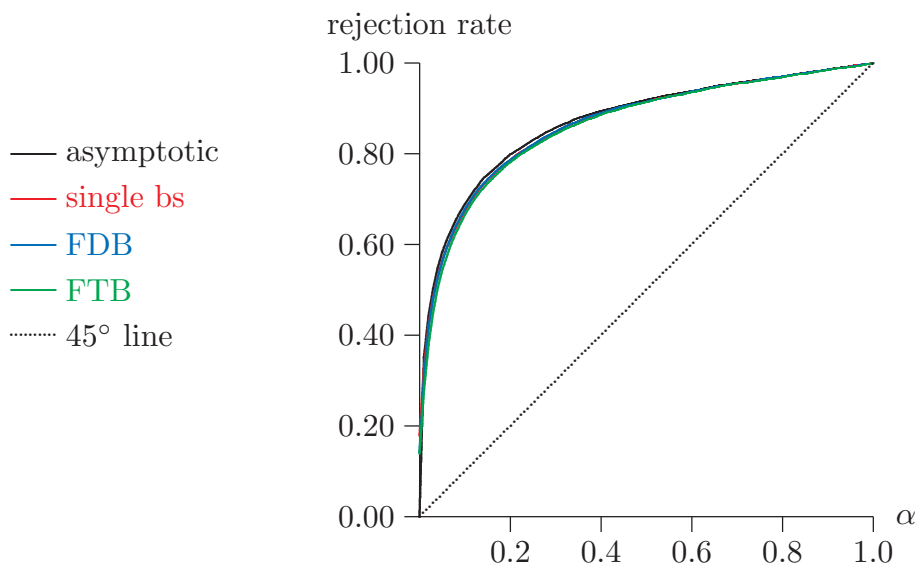


Figure 7: ERPs of Durbin-Godfrey test



$n=20$

$n=40$



$n=56$

Figure 8: Durbin-Godfrey test rejection rate, $\rho=0.5$