Size distortion of bootstrap tests: application to a unit root test

Russell Davidson

GREQAM

Groupement de Recherche en Economie
Quantitative d'Aix-Marseille - UMR-CNRS 6579
Ecole des Hautes Etudes en Sciences Sociales
Universités d'Aix-Marseille II et III

Document de Travail n°2009-40

SIZE DISTORTION OF BOOTSTRAP TESTS: APPLICATION TO A UNIT ROOT TEST

Russell DAVIDSON

August 2009

Size Distortion of Bootstrap Tests
Application to a Unit Root Test

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

Abstract

Testing for a unit root in a series obtained by summing a stationary MA(1) process with a parameter close to -1 leads to serious size distortions under the null, on account of the near cancellation of the unit root by the MA component in the driving stationary series. The situation is analysed from the point of view of bootstrap testing, and an exact quantitative account is given of the error in rejection probability of a bootstrap test. A particular method of estimating the MA parameter is recommended, as it leads to very little distortion even when the MA parameter is close to -1. A new bootstrap procedure with still better properties is proposed. While more computationally demanding than the usual bootstrap, it is much less so than the double bootstrap.

Keywords: Unit root test, bootstrap, MA(1), size distortion

JEL codes: C10, C12, C22

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. A first version of the paper was presented at the 2nd Annual Granger Centre Conference, on Bootstrap and Numerical Methods in Time Series. A second version was presented at the 4th London and Oxbridge Time Series Workshop, and at the Econometrics workshop held at Rimini in July 2009. I thank participants of all these conferences for useful comments.

August 2009

1. Introduction

There are well-known difficulties in testing for a unit root in a series obtained by summing a stationary series that is a moving average process with a parameter θ close to -1. Unless special precautions are taken, size distortions under the null lead to gross over-rejection of 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. It is natural to ask if the bootstrap can alleviate the problem. Since the null hypothesis is actually false when $\theta = -1$, we cannot expect much power when θ is close to -1, but we can hope to reduce size distortion.

Sieve bootstraps of one sort or another have been proposed for unit root testing when one wishes to be quite agnostic as to the nature of the driving process. One of the first papers to propose a sieve bootstrap is Bühlmann (1997). The idea was further developed in Bühlmann (1998), Choi and Hall (2000), and Park (2002). In Park (2003), it was shown that under certain conditions a sieve bootstrap test benefits from asymptotic refinements. The sieve in question is an AR sieve, whereby one seeks to model the stationary driving series by a finite-order AR process, the chosen order being data driven. Sieves based on a set of finite-order MA or ARMA processes are considered in Richard (2007b), and many of Park's results are shown to carry over to these sieve bootstraps. In particular, as might be expected, the MA sieve has better properties than the more usual AR sieve when the driving process is actually MA(1).

The purpose of this paper is to study the determinants of size distortion of bootstrap tests, and to look for ways to minimise it. Therefore, the problem considered in this paper is very specific, and as simple as possible.

- The unit root test on which the bootstrap tests are based is the augmented Dickey-Fuller test.
- It is supposed that it is known that the driving stationary process is MA(1), so that the only unknown quantity is the MA parameter.
- The bootstrap is a parametric bootstrap, for which it is assumed that the innovations of the MA(1) process are Gaussian.

Thus no sieve is used in the bootstrap procedure; the bootstrap samples are always drawn from an MA(1) process. The reason for concentrating on such a specific problem is that the bootstrap DGP is completely characterised by a single scalar parameter. This makes it possible to implement a number of procedures that are infeasible in more general contexts. I make no effort to use some of the testing procedures that minimise the size distortion, because the size distortion is the main focus of the analysis. In addition, no mention is made in the paper of asymptotic theory or asymptotic refinements, other than to mention that the asymptotic validity of bootstrap tests of the sort considered here has been established by Park (2003).

The fact that the null hypothesis is essentially one-dimensional, parametrised by the MA parameter, means that the parametric bootstrap can be analysed particularly simply. Because the bootstrap data-generating process (DGP) is completely determined by

one single parameter, it is possible to implement at small computational cost a theoretical formula for the bootstrap discrepancy, that is, the difference between the true rejection probability of a bootstrap test and the nominal significance level of the test. This makes it possible to estimate the bootstrap discrepancy much more cheaply than usual. Any procedure that gives an estimate of the rejection probability of a bootstrap test allows one to compute a corrected P value. This is just the estimated rejection probability for a bootstrap test at nominal level equal to the uncorrected bootstrap P value, that is, the estimated probability mass in the distribution of the bootstrap P value in the region more extreme than the realised P value.

Although it is straightforward to estimate the parameters of an AR(p) process by a linear regression, estimating the parameter(s) of an MA or ARMA process is much less simple. In Galbraith and Zinde-Walsh (1994) and (1997), estimators are proposed that are easy to compute, as they are based on running the sort of linear regression used for estimation of AR parameters. However, I show that their estimators are too inefficient for them to be used effectively in the bootstrap context when the MA parameter is close to -1. Although the maximum likelihood estimator is easy enough to program, I have found that computation time is much longer than for the Galbraith and Zinde-Walsh (GZW) techniques. Further, the MLE has the odd property that its distribution sometimes has an atom with positive probability located at the point where the parameter is exactly equal to -1. A bootstrap DGP with parameter equal to -1 violates a basic principle of bootstrapping, since such a DGP does not have a unit root, whereas the null hypothesis of a unit root test is that one does exist. Here, I propose estimators based on nonlinear least squares (NLS) that are faster to compute than the MLE, although slower than the GZW estimators. They seem almost as efficient as maximum likelihood, and have no atom at -1. It is shown that they work very well for bootstrapping.

In Section 2, the NLS estimators of the parameters of MA and ARMA processes are described, and given in specific detail for MA(1). In Section 3, the distribution of the NLS estimator for an MA(1) process is compared with those of the MLE and the GZW estimators in a set of simulation experiments. It is found that the GZW estimators are seriously biased and have large variance when the true MA parameter is close to -1. The NLS and ML estimators, on the other hand, are much less biased and dispersed, and resemble each other quite closely, except that the NLS estimator has no atom at -1. Then, in Section 4, the bootstrap discrepancy is studied theoretically, and shown to depend on the joint bivariate distribution of two random variables. In Section 5, simulation-based methods for estimating the bootstrap discrepancy, and approximations to the discrepancy, are studied and compared in another set of simulation experiments. Section 6 studies three possible corrected bootstrap tests: the double bootstrap of Beran (1988), the fast double bootstrap of Davidson and MacKinnon (2007), and a new bootstrap, dubbed the discrepancy-corrected bootstrap, that is a good deal less computationally intensive than the double bootstrap, although more so than the fast double bootstrap. It is seen to be at least as good as the other two corrected bootstraps. In Section 7, a possible way to correct the fast double bootstrap is discussed. Simulation experiments that investigate the power of the bootstrap test are presented in Section 8, and some concluding remarks are offered in Section 9.

2. Estimating ARMA models by Nonlinear Least Squares

Suppose that the times series u_t is generated by an ARMA(p, q) process, that we write as

$$(1 + \rho(\mathrm{L}))u_t = (1 + \theta(\mathrm{L}))\varepsilon_t, \tag{1}$$

where L is the lag operator, ρ and θ are polynomials of degree p and q respectively:

$$\rho(z) = \sum_{i=1}^{p} \rho_i z^i \quad \text{and} \quad \theta(z) = \sum_{j=1}^{q} \theta_j z^j.$$

Note that neither polynomial has a constant term. We wish to estimate the coefficients ρ_i, $i = 1, \ldots, p$, and θ_j, $j = 1, \ldots, q$, from an observed sample u_t, $t = 1, \ldots, n$, under the assumption that the series ε_t is white noise, with variance σ^2.

In a model to be estimated by least squares, the dependent variable, here u_t, is expressed as the sum of a regression function, which in this pure time-series case is a function of lags of u_t, and a white-noise disturbance. The disturbance is ε_t, and so we solve for it in (1) to get

$$\varepsilon = (1 + \theta(\mathrm{L}))^{-1}(1 + \rho(\mathrm{L}))u.$$

Here, we may omit the subscript t, and interpret u and ε as the whole series. Since ρ and θ have no constant term, the current value, u_t, appears on the right-hand side only once, with coefficient unity. Thus we have

$$\begin{aligned} \varepsilon &= u + \big((1 + \theta(\mathrm{L}))^{-1}(1 + \rho(\mathrm{L})) - 1\big)u \\ &= u + (1 + \theta(\mathrm{L}))^{-1}\big(1 + \rho(\mathrm{L}) - (1 + \theta(\mathrm{L}))\big)u. \end{aligned}$$

The nonlinear regression we use for estimation is then

$$u = (1 + \theta(\mathrm{L}))^{-1}(\theta(\mathrm{L}) - \rho(\mathrm{L}))u + \varepsilon. \tag{2}$$

Write $R(\mathrm{L}) = (1 + \theta(\mathrm{L}))^{-1}(\theta(\mathrm{L}) - \rho(\mathrm{L}))$. The regression function $R(\mathrm{L})u$ is a nonlinear function of the ARMA parameters, the ρ_i and the θ_j.

A good way to compute series like $R(\mathrm{L})u$ or its derivatives is to make use of the operation of convolution. For two series a and b, the convolution series c is defined by

$$c_t = \sum_{s=0}^{t} a_s b_{t-s}, \quad t = 0, 1, 2, \ldots. \tag{3}$$

The first observation is indexed by 0, because the definition is messier if the first index is 1 rather than 0. Convolution is symmetric in a and b and is linear with respect to each argument. In fact, the c_t are just the coefficients of the polynomial $c(z)$ given by the product $a(z)b(z)$, with $a(z) = \sum_t a_t z^t$, and similarly for $b(z)$ and $c(z)$.

Define the convolution operator C in the obvious way: $\mathrm{C}(a, b) = c$, where c is the series with c_t given by (3). Although the coefficients of the inverse of a polynomial can be computed using the binomial theorem, it is easier to define an inverse convolution function C^{-1} such that

$$a = \mathrm{C}^{-1}(c, b) \quad \text{iff} \quad c = \mathrm{C}(a, b).$$

Inverse convolution is *not* symmetric with respect to its arguments. It is linear with respect to its first argument, but not the second. It is easy to compute an inverse convolution recursively. If the relation is (3), then, if $b_0 = 1$ as is always the case here, we see that

$$a_0 = c_0 \quad \text{and} \quad a_t = c_t - \sum_{s=0}^{t-1} a_s b_{t-s}.$$

Note that $\mathrm{C}(a, e_0) = a$ and $\mathrm{C}^{-1}(a, e_0) = a$ where $(e_0)_t = \delta_{t0}$. This corresponds to the fact that the polynomial 1 is the multiplicative identity in the algebra of polynomials. In addition, if the series e_j is defined so as to have element t equal to δ_{tj}, then $\mathrm{C}(a, e_j) = \mathrm{L}^j a$.

Let a be the series the first element (element 0) of which is 1, element i of which is the AR parameter ρ_i, for $i = 1, \ldots, p$, and elements a_t for $t > p$ are zero. Define b similarly with the MA parameters. Then the series r containing the coefficients of $R(\mathrm{L})$ is given by

$$r = \mathrm{C}^{-1}(b - a, b),$$

and the series $R(\mathrm{L})u$ is just $\mathrm{C}(u, r)$. The derivatives of $R(\mathrm{L})u$ with respect to the ρ_i and the θ_j are also easy to calculate.

MA(1)

It is useful to specialise the above results for the case of an MA(1) process. The series a is then e_0, and b is $e_0 + \theta e_1$, where we write θ instead of θ_1, since there are no other parameters. Then $R(\mathrm{L}) = \theta(1 + \theta\mathrm{L})^{-1}\mathrm{L}$, and the coefficients of the polynomial R are the elements of the series $r = R(\mathrm{L})e_0 = \theta\mathrm{C}^{-1}(\mathrm{L}e_0, b)$. Consequently, $\mathrm{C}(u, r) = \theta\mathrm{C}^{-1}(\mathrm{L}u, b)$. The only derivative of interest is with respect to θ; it is $\mathrm{C}^{-1}\big(\mathrm{L}(u - \mathrm{C}(u, r)), b\big)$.

The regression (2), which we write as $u = R(\mathrm{L})u + \varepsilon$, is not in fact accurate for a finite sample, because the convolution operation implicitly sets the elements with negative indices of all series equal to 0. For the first element, the regression says therefore that $u_0 = \varepsilon_0$, whereas what should be true is rather that $u_0 = \varepsilon_0 + \theta\varepsilon_{-1}$. Thus the relation $u - \theta\varepsilon_{-1}e_0 = (1 + \theta\mathrm{L})\varepsilon$ is true for all its elements if the lag of ε_0 is treated as zero. We write $\phi = \theta\varepsilon_{-1}$, and treat ϕ as an unknown parameter. The regression model (2) is replaced by

$$u = \phi e_0 + \theta(1 + \theta\mathrm{L})^{-1}\mathrm{L}(u - \phi e_0) + \varepsilon. \tag{4}$$

Although it is perfectly possible to estimate (4) by nonlinear least squares, with two parameters, θ and ϕ, it is faster to perform two nonlinear regressions, each with only one parameter θ. When there is only one parameter, the least-squares problem can be solved as a one-dimensional minimisation. The first stage sets $\phi = 0$ in order to get a

preliminary estimate of θ; then, for the second stage, ϕ is estimated from the first-stage result, and the result used as a constant in the second stage. The first-order condition for ϕ in the regression (4) is

$$\big((1 - R(\mathrm{L})e_0\big)^\top\big((1 - R(\mathrm{L})\big)(u - \phi e_0)\big) = 0.$$

Recalling that $R(\mathrm{L})e_0 = r$ and writing $e_0 - r = s$, we can write this condition as

$$s^\top\big((1 - R(\mathrm{L}))u - \phi s\big) = 0 \quad \text{whence} \quad \phi = \frac{s^\top(1 - R(\mathrm{L}))u}{s^\top s}.$$

In order to compute the estimate of ϕ from the first stage, the series s is set up with $s_0 = 1$, $s_t = -r_t$ for $t > 0$, and we note that $(1 - R(\mathrm{L}))u$ is just the vector of residuals from the first stage regression.

3. Comparison of Estimators for MA(1)

Asymptotic efficiency in the estimation of the parameter θ of an MA(1) process is achieved by Gaussian maximum likelihood (ML) if the disturbances are Gaussian. But we saw that the MLE has an atom at -1 if the true θ is close to -1, and so it is of interest to see how much efficiency is lost by using other methods that do not have this feature.

The loglikelihood function for the MA(1) model is

$$\ell(\theta, \sigma^2) = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2}\log\det \boldsymbol{\Sigma}(\theta) - \frac{1}{2\sigma^2}\boldsymbol{u}^\top\boldsymbol{\Sigma}^{-1}(\theta)\boldsymbol{u}, \tag{5}$$

where $\sigma^2 = \mathrm{Var}(\varepsilon_t)$ and $\boldsymbol{\Sigma}(\theta)$ is an $n \times n$ Toeplitz matrix with all diagonal elements equal to $1 + \theta^2$ and all elements of the diagonals immediately below and above the principal diagonal equal to θ. The notation $\boldsymbol{u}$ is just vector notation for the series u.

Concentrating with respect to σ^2 gives

$$\hat{\sigma}^2(\theta) = \frac{1}{n}\boldsymbol{u}^\top\boldsymbol{\Sigma}^{-1}(\theta)\boldsymbol{u}.$$

Thus the concentrated loglikelihood is

$$\frac{n}{2}(\log n - \log 2\pi - 1) - \frac{n}{2}\log \boldsymbol{u}^\top\boldsymbol{\Sigma}^{-1}(\theta)\boldsymbol{u} - \frac{1}{2}\log\det \boldsymbol{\Sigma}(\theta). \tag{6}$$

This expression can be maximised with respect to θ by minimising

$$\ell(\theta) \equiv n\log \boldsymbol{u}^\top\boldsymbol{\Sigma}^{-1}(\theta)\boldsymbol{u} + \log\det \boldsymbol{\Sigma}(\theta), \tag{7}$$

and this can be achieved by use of any suitable one-dimensional minimisation algorithm, including Newton's method.

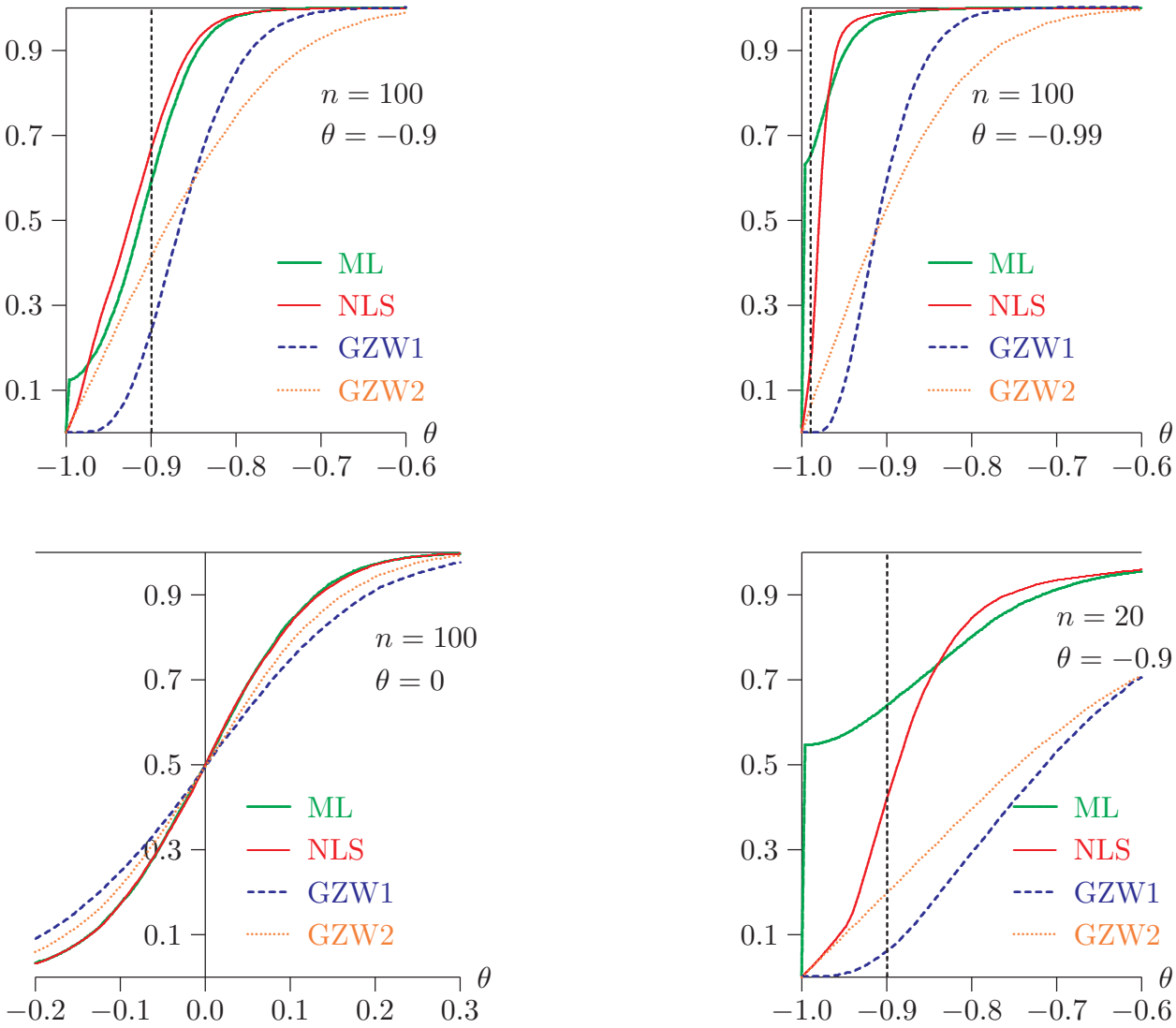

Figure 1: Comparison of MA(1) estimators

Other methods for estimating MA models have been proposed by Galbraith and Zinde-Walsh (1994) and for ARMA models by the same authors (1997). Their methods are based on estimating the following AR(k) model by ordinary least squares:

$$u_t = \sum_{i=1}^{k} a_i u_{t-i} + \text{residual}, \quad t = k+1, \ldots, n. \tag{8}$$

For an ARMA(p, q) model, k is chosen considerably greater than $p + q$. The estimators are consistent as $k \to \infty$ while $k/n \to 0$, but are not asymptotically efficient. They are however simple and fast to compute, as they involve no iterative procedure.

Let the OLS estimates of the parameters a_i in (8) be denoted as $\hat{a}_i$. For the MA(1) model $u_t = \varepsilon_t + \theta\varepsilon_{t-1}$, the simplest estimator of θ is just $\hat{a}_1$. Another estimator, that can be

traced back to Durbin (1959), is the parameter estimate from the OLS regression of the vector $[\hat{a}_1 \dots \hat{a}_k]^\top$, on the vector $[1 \quad -\hat{a}_1 \dots -\hat{a}_{k-1}]^\top$.

Any of the estimation methods so far discussed can give an estimate of θ outside the interval $[-1, 1]$. But the processes with parameters θ and $1/\theta$ are observationally equivalent. Thus whenever an estimate outside $[-1, 1]$ is obtained, it is simply replaced by its reciprocal. In Figure 1 are shown estimated cumulative distribution functions (CDFs) of four estimators, (Gaussian) maximum likelihood (ML), nonlinear least-squares using the two-stage procedure based on (4), with $\rho(z) = 0$, $\theta(z) = \theta z$ (NLS), Galbraith and Zinde-Walsh's first estimator $\hat{a}_1$ (GZW1), and their second estimator (GZW2). In all but the bottom right panel of the figure the sample size is $n = 100$. The distributions are shown for values of θ of -0.9, -0.99, and 0. The length of the GZW preliminary autoregression (8) is $k = 20$. In the bottom right panel, $n = 20$, $\theta = -0.9$, and $k = 6$. It is well known that the greatest challenge for estimators of the MA(1) parameter arises when θ is close to -1. The overall picture is clear enough. Both ML and NLS outperform the GZW estimators except when $\theta = 0$, or, more generally, when θ is distant from -1. GZW1 has much greater variance than the other estimators, and GZW2 is heavily biased to the right. For $n = 20$, the concentration of ML estimates close to -1 is seen; the other estimators do not exhibit this feature, which is much less visible for ML itself for the larger sample size. ML and NLS are almost unbiased for $n = 100$, and do not greatly differ. Experiments with other values of θ show that the four estimators have similar distributions when n is large enough and θ is greater than around -0.5 The inescapable conclusion is that using the GZW estimator in order to define a bootstrap DGP will give rise to serious size distortion.

4. The Bootstrap Discrepancy

Suppose that a test statistic τ is designed to test a particular null hypothesis. The set of all DGPs that satisfy that hypothesis is denoted as $\mathbb{M}_0$; this set constitutes what we may call the null model. A bootstrap test based on the statistic τ approximates the distribution of τ under a DGP $\mu \in \mathbb{M}_0$ by its distribution under a bootstrap DGP that also belongs to $\mathbb{M}_0$ and can be thought of as an estimate of the true DGP μ.

We define the bootstrap discrepancy as the difference, as a function of the true DGP and the nominal level, between the actual rejection probability of the bootstrap test and the nominal level. In order to study it, we suppose, without loss of generality, that the test statistic is already in approximate P value form, so that the rejection region is to the left of a critical value.

The rejection probability function R depends both on the nominal level α and the DGP μ. It is defined as

$$R(\alpha, \mu) \equiv \Pr_\mu(\tau < \alpha). \tag{9}$$

We assume that, for all $\mu \in \mathbb{M}_0$, the distribution of τ has support $[0, 1]$ and is absolutely continuous with respect to the uniform distribution on that interval. For given μ, $R(\alpha, \mu)$ is just the CDF of τ evaluated at α. The inverse of the rejection probability function is the critical value function Q, which is defined implicitly by the equation

$$\Pr_\mu\big(\tau < Q(\alpha, \mu)\big) = \alpha. \tag{10}$$

It is clear from (10) that $Q(\alpha, \mu)$ is the α-quantile of the distribution of τ under μ. In addition, the definitions (9) and (10) imply that

$$R\big(Q(\alpha, \mu), \mu\big) = Q\big(R(\alpha, \mu), \mu\big) = \alpha \tag{11}$$

for all α and μ.

In what follows, we ignore simulation randomness in the estimate of the distribution of τ under the bootstrap DGP, which we denote by μ^*. The bootstrap critical value for τ at level α is $Q(\alpha, \mu^*)$. Rejection by the bootstrap test is the event $\tau < Q(\alpha, \mu^*)$. Applying the increasing transformation $R(\cdot, \mu^*)$ to both sides and using (11), we see that the bootstrap test rejects whenever

$$R(\tau, \mu^*) < R\big(Q(\alpha, \mu^*), \mu^*\big) = \alpha.$$

Thus the bootstrap P value is just $R(\tau, \mu^*)$, which can therefore be interpreted as a bootstrap test statistic.

If instead we apply the increasing transformation $R(\cdot, \mu)$ to the inequality $\tau < Q(\alpha, \mu^*)$, it follows that rejection by the bootstrap test can also be expressed as $R(\tau, \mu) < R\big(Q(\alpha, \mu^*), \mu\big)$. We define two random variables, one a deterministic function of τ, the other a deterministic function of μ^*, the other random element involved in the bootstrap test. The first variable is $p \equiv R(\tau, \mu)$. It is distributed as U(0,1) under μ, because $R(\cdot, \mu)$ is the CDF of τ under μ and because we have assumed that the distribution of τ is absolutely continuous on the unit interval for all $\mu \in \mathbb{M}$. The second random variable is $q \equiv R(Q(\alpha, \mu^*), \mu) - \alpha = R(Q(\alpha, \mu^*), \mu) - R(Q(\alpha, \mu), \mu)$. Thus rejection by the bootstrap test is the event $p < \alpha + q$. Let the CDF of q under μ conditional on the random variable p be denoted as $F(q \mid p)$. Then it is shown in Davidson and MacKinnon (2006) that the bootstrap discrepancy can be expressed as

$$\int_{-\alpha}^{1-\alpha} x \, \mathrm{d}F(x \mid \alpha + x). \tag{12}$$

The random variable $q + \alpha$ is the probability that a statistic generated by the DGP μ is less than the α-quantile of the bootstrap distribution, conditional on that distribution. The expectation of q can thus be interpreted as the bias in rejection probability when the latter is estimated by the bootstrap. The actual bootstrap discrepancy, which is a nonrandom quantity, is the expectation of q conditional on being at the margin of rejection.

We study the critical value function of the test statistic most frequently used to test the null hypothesis of a unit root, namely the augmented Dickey-Fuller (ADF) test. The DGPs used to generate the data of the simulation experiment take the form

$$u_t = \varepsilon_t + \theta \varepsilon_{t-1}, \quad y_t = y_0 + \sum_{s=1}^{t} u_t, \tag{13}$$

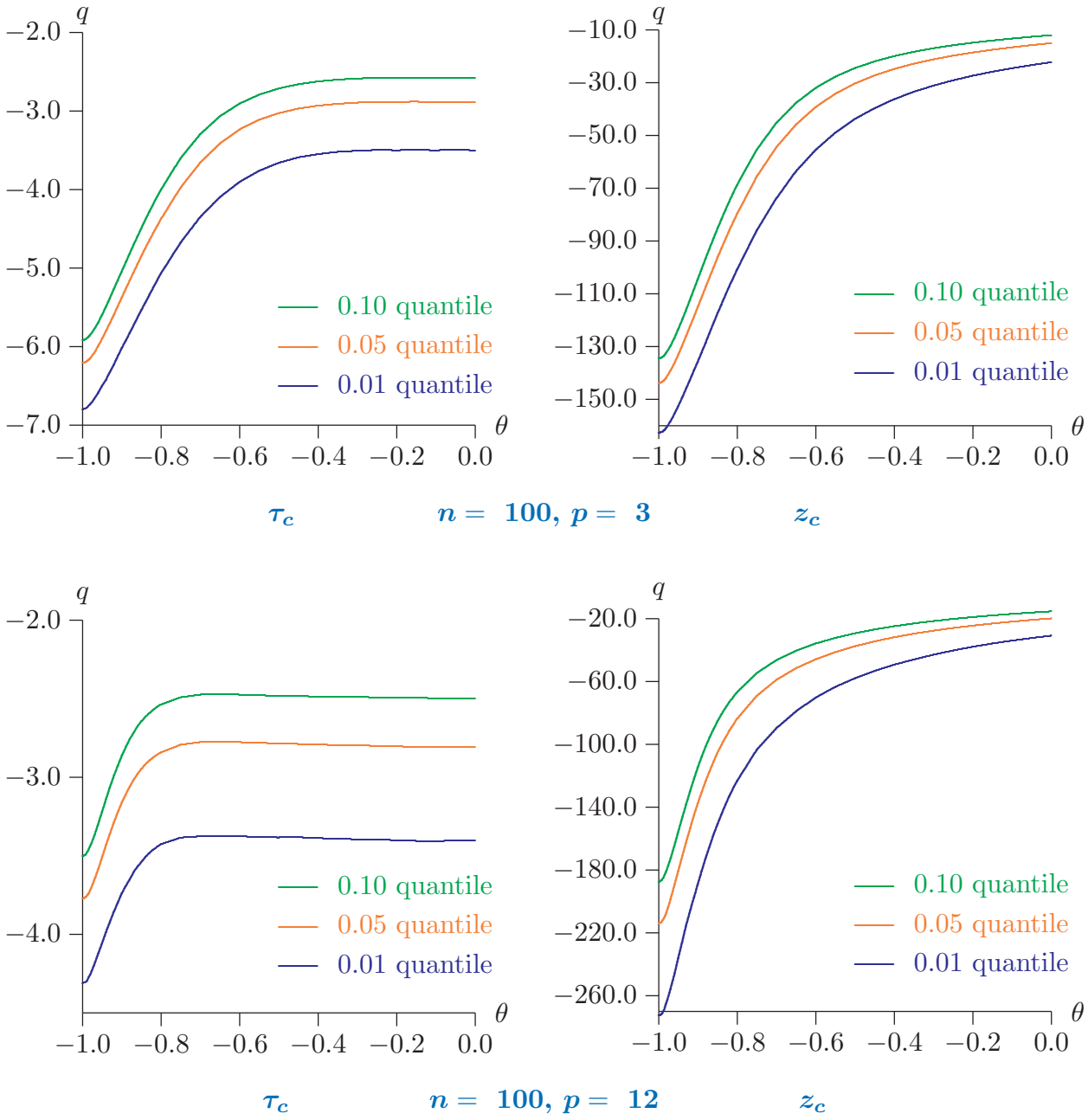

Figure 2: Critical value functions

where the ε_t are IID N(0,1). The test statistics are computed using the ADF testing regression

$$\Delta y_t = \beta_0 + \beta_1 y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-1} + \text{residual}. \tag{14}$$

When this regression is run by ordinary least squares, the z_c statistic is $n\hat{\beta}_1$; the τ_c statistic is the conventional t statistic for the hypothesis that $\beta_1 = 0$. Under the null hypothesis that the series y_t has a unit root, these two statistics have well-known but nonstandard

asymptotic distributions.

The variance of the ε_t is set to 1 without loss of generality, since both statistics are scale invariant. In fact, both statistics are also numerically invariant to changes in the value of the starting value y_0, and so we can without loss of generality set $y_0 = 0$ in our simulations. We vary the MA parameter θ from 0 to -0.8 by steps of 0.05, and from -0.8 to -0.99 by steps of 0.01. For each value we estimate the 0.01, 0.05, and 0.10 quantiles of the distribution of each statistic, using 99,999 replications. The same random numbers are used for each value of θ in order to achieve a smoother estimate of the critical value function, which is the quantile of the statistic as a function of θ.

Figure 2 shows the 0.01, 0.05, and 0.1 quantiles for the τ_c and z_c statistics. We could obtain the CVFs of the statistics by transforming them by the inverse of the nominal asymptotic CDFs of the statistics. In the upper panels, the sample size $n = 100$ and the number p of lags of Δy_t is 3. In the lower panels, the quantiles are graphed for $n = 100$ and $p = 12$. The choice of the statistic τ_c and of $p = 12$ gives the smallest variation of the CVF, and so, for the rest of this study, we examine the consequences of making this choice.

5. Estimating the Bootstrap Discrepancy

Brute force

The conventional way to estimate the bootstrap rejection probability (RP) for a given DGP μ and sample size n by simulation is to generate a large number, M say, of samples of size n using the DGP μ. For each replication, a realization τ_m of the statistic τ is computed from the simulated sample, along with a realization $\hat{\mu}_m$ of the bootstrap DGP. Then B bootstrap samples are generated using $\hat{\mu}_m$, and bootstrap statistics τ^*_{mj}, $j = 1, \ldots, B$ are computed. The realized bootstrap P value for replication m is then

$$\hat{p}^*_m(\tau_m) \equiv \frac{1}{B} \sum_{j=1}^{B} \mathrm{I}(\tau^*_{mj} < \tau_m), \tag{15}$$

where we assume that the rejection region is to the left. The estimate of the RP at nominal level α is the proportion of the $\hat{p}^*_m(\tau_m)$ that are less than α. The whole procedure requires the computation of $M(B + 1)$ statistics and M bootstrap DGPs. The bootstrap statistics τ^*_{mj} are realizations of a random variable that we denote as τ^*.

If one wishes to compare the RP of the bootstrap test with that of the underlying asymptotic test, a simulation estimate of the latter can be obtained directly as the proportion of the τ_m less than the asymptotic α level critical value. Of course, estimation of the RP of the asymptotic test by itself requires the computation of only M statistics.

Denote by p^*_1 the ideal bootstrap P value, that is, the probability mass in the distribution of the bootstrap statistics in the region more extreme than the realisation $\hat{\tau}$ of the statistic computed from the real data. Let the probability space in which statistics and bootstrap DGPs are defined be $(\Omega, \mathcal{F}, P)$. The statistic $\hat{\tau}$ can then be written as $\tau(\omega, \mu)$, where

$\omega \in \Omega$ and μ is the true DGP. The bootstrap DGP can be expressed as $\mu(\omega, \mu)$, for the same realisation ω as for $\hat{\tau}$. In a simulation context, the probability space can be considered that of the random number generator. With real data, ω is just a way of representing all the random elements that gave rise to the data.

The bootstrap P value can be expressed as

$$p_1^*(\omega, \mu) = \Pr_\mu(\tau^* < \hat{\tau}) = \mathrm{E}_\mu\big(\mathrm{I}(\tau^* < \hat{\tau}) \,|\, \omega\big),$$

where τ^* is the bootstrap statistic. A realisation of τ^* is $\tau\big(\omega^*, \mu(\omega, \mu)\big)$, where $\omega^* \in \Omega$ is independent of ω. The P value is thus

$$p_1^*(\omega, \mu) = \int_\Omega \mathrm{I}\big(\tau(\omega^*, \mu(\omega, \mu)) < \tau(\omega, \mu)\big) \,\mathrm{d}P(\omega^*). \tag{16}$$

Let $R(x, \mu)$ be the CDF of τ under μ. This means that

$$R(x, \mu) = \Pr\big(\tau(\omega, \mu) < x\big) = \int_\Omega \mathrm{I}\big(\tau(\omega, \mu) < x\big) \,\mathrm{d}P(\omega),$$

and so, from (16),

$$p_1^*(\omega, \mu) = R\big(\tau(\omega, \mu), \mu(\omega, \mu)\big).$$

We denote the CDF of this random variable by $R_1(x, \mu)$, so that

$$R_1(x, \mu) = \Pr_\mu(p_1^*(\mu, \omega) \le x) = \mathrm{E}_\mu\Big(\mathrm{I}\big(R(\tau(\mu, \omega), b(\mu, \omega)) \le x\big)\Big). \tag{17}$$

The fast approximation

It is shown in Davidson and MacKinnon (2007) that, under certain conditions, it is possible to obtain a much less expensive approximate estimate of the bootstrap RP, as follows. As before, for $m = 1, \dots, M$, the DGP μ is used to draw realizations τ_m and $\hat{\mu}_m$. In addition, $\hat{\mu}_m$ is used to draw a *single* bootstrap statistic τ_m^*. The τ_m^* are therefore IID realizations of the variable τ^*. We estimate the RP as the proportion of the τ_m that are less than $\hat{Q}^*(\alpha)$, the α quantile of the τ_m^*. This yields the following estimate of the RP of the bootstrap test:

$$\widehat{\mathrm{RP}}_\mathrm{A} \equiv \frac{1}{M} \sum_{m=1}^{M} \mathrm{I}\big(\tau_m < \hat{Q}^*(\alpha)\big),$$

As a function of α, $\widehat{\mathrm{RP}}_\mathrm{A}$ is an estimate of the CDF of the bootstrap P value.

The above estimate is approximate not only because it rests on the assumption of the full independence of τ and μ^*, but also because its limit as $B \to \infty$ is not precisely the RP of the bootstrap test. Its limit differs from the RP by an amount of a smaller order of magnitude than the difference between the RP and the nominal level α. But it requires the computation of only $2M$ statistics and M bootstrap DGPs.

Conditional on the bootstrap DGP μ^*, the CDF of τ^* evaluated at x is $R(x, \mu^*)$. Therefore, if μ^* is generated by the DGP μ, the unconditional CDF of τ^* is

$$R^*(x, \mu) \equiv \mathrm{E}_\mu\big(R(x, \mu^*)\big).$$

We denote the α quantile of the distribution of τ^* under μ by $Q^*(\alpha, \mu)$. In the explicit notation used earlier, since $\tau^* = \tau\big(\omega^*, \mu(\omega, \mu)\big)$, we see that

$$R^*(x, \mu) = \int_\Omega \int_\Omega \mathrm{I}\Big(\tau\big(\omega^*, \mu(\omega, \mu)\big) < x\Big)\, \mathrm{d}P(\omega^*)\, \mathrm{d}P(\omega). \tag{18}$$

Evaluating the analytic expression

The formula (12) cannot be implemented unless one knows the function F, the CDF of q conditional on p. This function can be estimated arbitrarily well by simulation if we can generate IID joint realisations of p and q. But that is made difficult by the fact that, for a given DGP μ, both p and q are defined in terms of the functions R and Q, which are in general unknown. Estimating R and Q by simulation is also quite possible, for a given μ. But q is defined using the bootstrap DGP μ^*, and, since this is random, we cannot estimate $Q(\cdot, \mu^*)$ for all possible realisations of μ^* in a single experiment.

The case of the model with DGPs of the form (13) is much more tractable than most, however, since the bootstrap DGP is completely determined by a single parameter. It is therefore convenient to replace the notation μ by θ. Simulations of the sort used to obtain the data graphed in Figure 2 can be used for simulation-based estimates of $Q(\alpha, \theta)$ for any given α and θ. For a set of values of α and a set of values of θ, we can construct a table giving the values of $Q(\alpha, \theta)$ for the chosen arguments. There are as many experiments as there are values of θ, but, as for Figure 3, it is advisable to use the same random numbers for each experiment. Each experiment allows us to estimate all the quantiles $Q(\alpha, \theta)$ for the relevant θ.

The most direct way to proceed after setting up the table is as follows. Choose a DGP by specifying the parameter θ, and choose a nominal level α. Use the chosen DGP to generate many joint realisations of the pair $(\tau, \hat{\theta})$. Approximate $Q(\alpha, \hat{\theta})$ by interpolation based on the values in the table for the chosen α and the set of θ values, obtaining the approximation $\tilde{Q}(\alpha, \hat{\theta})$. Then the rejection probability of the bootstrap test at level α is estimated by the proportion of realisations for which $\tau < \tilde{Q}(\alpha, \hat{\theta})$.

Alternatively, an experiment that more closely mimics the theoretical discussion leading to (12) is as follows. Perform a set of experiments in which we obtain simulation-based estimates of $R(Q(\alpha, \theta_1), \theta_2)$ for the set of values chosen for α, and for any pair (θ_1, θ_2) of values in the set chosen for θ. Here we need only as many experiments as there are values of θ, since, after fixing θ_2, we generate a large number of τ statistics using the DGP characterised by θ_2, and then, for each value of $Q(\alpha, \theta_1)$ in the set, estimate $R(Q(\alpha, \theta_1), \theta_2)$ as the proportion of the generated statistics less than $Q(\alpha, \theta_1)$.

Now suppose that a single realisation from the DGP characterised by a value of θ in the chosen set gives rise to an estimate $\hat{\theta}$. For given α, we can then use the simulated values of $R(Q(\alpha, \theta_1), \theta)$ in order to interpolate the value of $R(Q(\alpha, \hat{\theta}), \theta)$. As Figure 3 shows, the quantiles vary quite smoothly as functions of θ, and so interpolation should work well. In the experiments to be described, cubic splines were used for this purpose.

If we now repeat the operation of the previous paragraph many times, we get a set of realisations of the random variable q by subtracting α from the simulated $R(Q(\alpha, \hat{\theta}), \theta)$. But for each repetition, we also compute the value of the τ statistic, and keep the pair (τ, q). When all the repetitions have been completed, we sort the pairs in increasing order of τ. For each repetition, then, we estimate the random variable p by the index of the associated pair in the sorted set, divided by the number of repetitions. This is equivalent to using the set of generated τ values to estimate $R(\cdot, \theta)$, and evaluating the result at the particular τ for each repetition. We end up with a set of IID joint realisations of p and q.

At this point, our estimate of the RP of the bootstrap test at significance level α is just the proportion of the repetitions for which $p < \alpha + q$, and the estimate of the bootstrap discrepancy is the estimated RP minus α. It is of interest to see how close two approximations to the bootstrap discrepancy come to the estimate obtained in this way. Both of these can be readily computed using the set of joint realisations of p and q. The first is the estimated expectation of q, which is just the average of the realised q, with no reference to the associated p. The second is an estimate of the expectation of q conditional on $p = \alpha$, that is,

$$\int_{-\alpha}^{1-\alpha} x \, \mathrm{d}F(x \,|\, \alpha),$$

rather than the exact expression (12). This conditional expectation can readily be estimated with a kernel estimator.

In Figure 3 are depicted plots of the bootstrap discrepancy as a function of the nominal level α for values between 0.01 and 0.10. The three panels show results for three different true DGPs, with $\theta = 0.90$, 0.95, and 0.99, with sample size $n = 100$. The NLS procedure was used for the estimation of θ. Three simulation-based estimates are given. The first two were computed using 99,999 repetitions of the experiment described above, the first the proportion of repetitions with $p < \alpha + q$, the second the expectation of q. The expectation conditional on $p = \alpha$ is so close to the first estimate that it would not be distinguishable from it in the graph. The last estimate was computed after 10,000 repetitions of a full-blown bootstrap test, with 399 bootstrap repetitions.

It can be seen that the unconditional expectation of q is not a very good estimate of the bootstrap discrepancy. In all cases, it overestimates the RP. Of the other two estimates, the one based on the realisations of p and q is probably superior from the theoretical point of view, since the one based on full-blown bootstrapping, besides being based on fewer repetitions, gives the bootstrap discrepancy for a test *with* 399 *bootstrap repetitions*, while the other estimates the theoretical bootstrap discrepancy, corresponding to an infinite number of bootstrap repetitions. An interesting inversion of the sign of the bootstrap discrepancy can be seen, with a negative discrepancy for both $\theta = -0.90$ and $\theta = -0.95$,

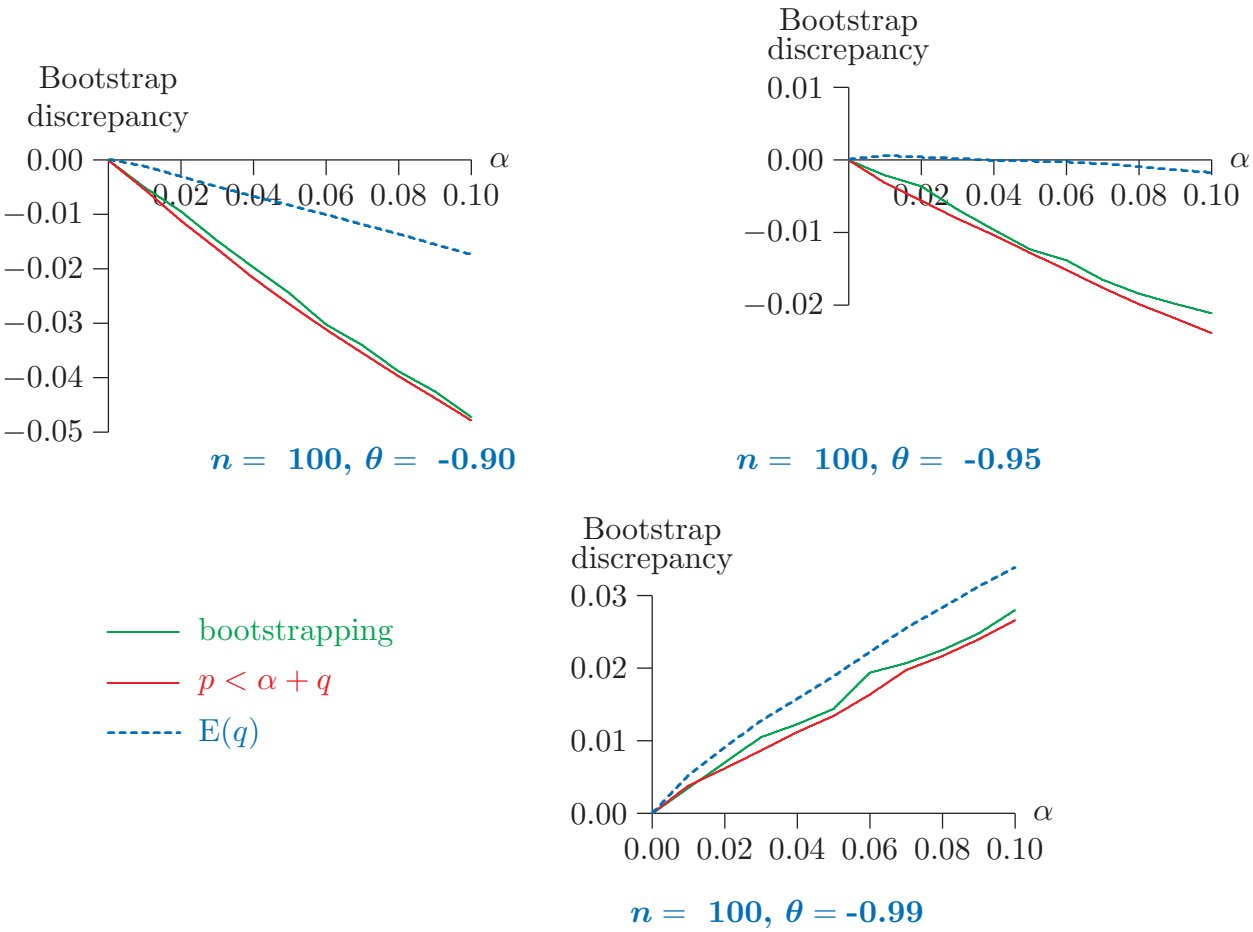

Figure 3: Bootstrap discrepancy; NLS estimation

but positive for $\theta = -0.99$. This last phenomenon is expected, since the asymptotic ADF test overrejects grossly for θ close to -1. However, even for $\theta = -0.95$, the discrepancy is negative. Note also that the bootstrap discrepancy is nothing like as large as the error in rejection probability of the asymptotic test, and, even for $\theta = -0.99$, is just over 1% for a nominal level of 5%.

In Figure 4, results like those in Figure 3 are shown when θ is estimated using the GZW2 estimator. A fourth curve is plotted, giving the estimate based on the expectation of q conditional on $p = \alpha$. It is no longer indistinguishable from the estimate based on the frequency of the event $p < \alpha + q$. This latter estimate, on the other hand, is very close to the one based on actual bootstrapping. Overall, the picture is very different from what we see with the NLS estimator for θ. The overrejection of the asymptotic test reappears for all three values of θ considered, and, although it is less severe, it is still much too great for $\theta = -0.95$ and $\theta = -0.99$ for the test to be of any practical use. Again, the unconditional expectation of q overestimates the RP. Evidently, bootstrap performance is much degraded by the use of the less efficient estimator of θ. The results in Figure 4 are much more similar to those in Richard (2007b) than are those of Figure 3.

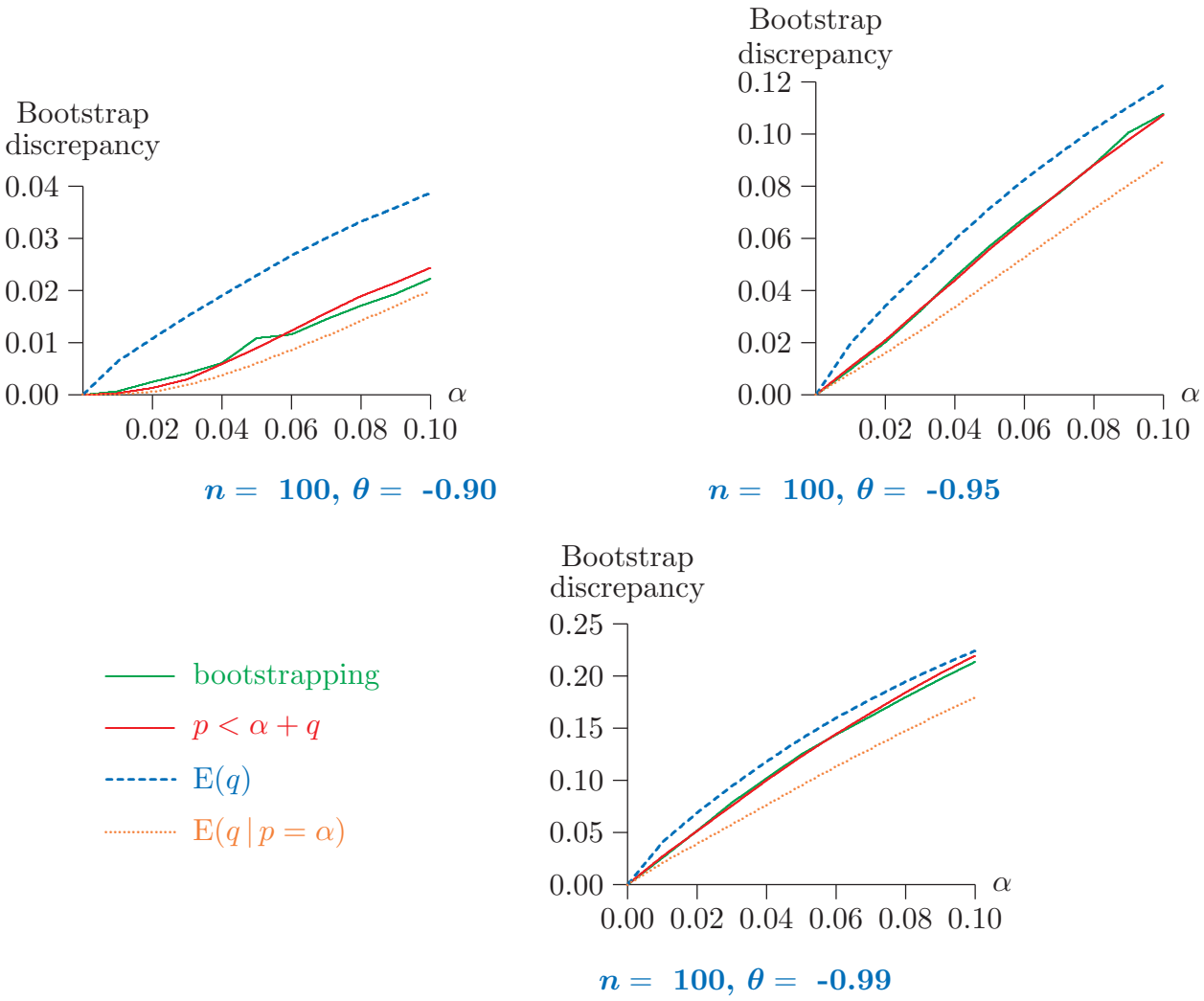

Figure 4: Bootstrap discrepancy; GZW2 estimation

6. Bootstrapping the Bootstrap Discrepancy

Any procedure that gives an estimate of the rejection probability of a bootstrap test, or of the CDF of the bootstrap P value, allows one to compute a corrected P value. In principle, the analysis of the previous section gives the CDF of the bootstrap P value, and so it is interesting to see if we can devise a way to exploit this, and compare it with two other techniques sometimes used to obtain a corrected bootstrap P value, namely the double bootstrap, as originally proposed by Beran (1988), and the fast double bootstrap proposed by Davidson and MacKinnon (2007).

The double bootstrap

An estimate of the bootstrap RP or the bootstrap discrepancy is specific to the DGP that generates the data. Thus what is in fact done by all techniques that aim to correct a bootstrap P value is to *bootstrap* the estimate of the bootstrap RP, in the sense that the

bootstrap DGP itself is used to estimate the bootstrap discrepancy. This can be seen for the ordinary double bootstrap as follows.

The brute force method described earlier for estimating the RP of the bootstrap test is employed, but with the (first-level) bootstrap DGP in place of μ. The first step is to compute the usual bootstrap P value, p_1^* say, using B_1 bootstrap samples generated from a bootstrap DGP μ^*. Now one wants an estimate of the actual RP of a bootstrap test at nominal level p_1^*. This estimated RP is the double bootstrap P value, p_2^{**}. Thus we set $\mu = \mu^*$, $M = B_1$, and $B = B_2$ in the brute-force algorithm described in the previous section. The computation of p_1^* has already provided us with B_1 statistics τ_j^*, $j = 1, \ldots, B_1$, corresponding to the τ_m of the algorithm. For each of these, we compute the (double) bootstrap DGP μ_j^{**} realised jointly with τ_j^*. Then μ_j^{**} is used to generate B_2 second-level statistics, which we denote by τ_{jl}^{**}, $l = 1, \ldots, B_2$; these correspond to the τ_{mj}^* of the algorithm. The second-level bootstrap P value is then computed as

$$p_j^{**} = \frac{1}{B_2} \sum_{l=1}^{B_2} \mathrm{I}(\tau_{jl}^{**} < \tau_j^*); \tag{19}$$

compare (15). The estimate of the bootstrap RP at nominal level p_1^* is then the proportion of the p_j^{**} that are less than p_1^*:

$$p_2^{**} = \frac{1}{B_1} \sum_{j=1}^{B_1} \mathrm{I}\big(p_j^{**} \leq p_1^*\big). \tag{20}$$

The inequality in (20) is not strict, because there may well be cases for which $p_j^{**} = p_1^*$. For this reason, it is desirable that $B_2 \neq B_1$. The whole procedure requires the computation of $B_1(B_2 + 1) + 1$ statistics and $B_1 + 1$ bootstrap DGPs.

Recall from (17) that $R_1(x, \mu)$ is our notation for the CDF of the first-level bootstrap P value. The double bootstrap P value is thus

$$p_2^{**}(\omega, \mu) \equiv R_1\big(p_1^*(\omega, \mu), \mu(\omega, \mu)\big) = R_1\big(R(\tau(\omega, \mu), \mu(\omega, \mu)), \mu(\omega, \mu)\big).$$

The fast double bootstrap

The so-called fast double bootstrap (FDB) of Davidson and MacKinnon (2007) is much less computationally demanding than the double bootstrap, being based on the fast approximation of the previous section. Like the double bootstrap, the FDB begins by computing the usual bootstrap P value p_1^*. In order to obtain the estimate of the RP of the bootstrap test at nominal level p_1^*, we use the algorithm of the fast approximation with $M = B$ and $\mu = \mu^*$. For each of the B samples drawn from μ^*, we obtain the ordinary bootstrap statistic τ_j^*, $j = 1, \ldots, B$, and the double bootstrap DGP μ_j^{**}, exactly as with the double bootstrap. One statistic τ_j^{**} is then generated by μ_j^{**}. The p_1^* quantile of the τ_j^{**}, say $Q^{**}(p_1^*)$, is then computed. Of course, for finite B, there is a range of values that can

be considered to be the relevant quantile, and we must choose one of them somewhat arbitrarily. The FDB P value is then

$$p^*_{\mathrm{FDB}} = \frac{1}{B}\sum_{j=1}^{B} \mathrm{I}\big(\tau^*_j < Q^{**}(p^*_1)\big).$$

To obtain it, we must compute $2B+1$ statistics and $B+1$ bootstrap DGPs.

In explicit notation, we have

$$p^*_{\mathrm{FDB}}(\omega,\mu) = \Pr\Big(\tau^* < Q^*\big(p^*_1(\omega,\mu),\mu(\omega,\mu)\big) \;\Big|\; \omega\Big).$$

where $Q^*(\cdot,\mu)$ is the quantile function corresponding to the CDF $R^*(x,\mu)$ of (18).

More explicitly still, we have that

$$\begin{aligned} p^*_{\mathrm{FDB}}(\omega,\mu) &= \int_\Omega \mathrm{I}\Big(\tau\big(\omega^*,\mu(\omega,\mu)\big) < Q^*\big(p^*_1(\omega,\mu),\mu(\omega,\mu)\big)\Big)\,\mathrm{d}P(\omega^*) \\ &= R\Big(Q^*\big(p^*_1(\omega,\mu),\mu(\omega,\mu)\big),\mu(\omega,\mu)\Big) \\ &= R\Big(Q^*\big(R(\tau(\omega,\mu),\mu(\omega,\mu)),\mu(\omega,\mu)\big),\mu(\omega,\mu)\Big). \end{aligned} \tag{21}$$

The discrepancy-corrected bootstrap

What makes the technique of the previous section for estimating the bootstrap discrepancy computationally intensive is the need to set up the tables giving $Q(\alpha,\theta)$ for a variety of values of α and θ. For a fixed α, of course, we need only vary θ, and this is the state of affairs when we wish to correct a bootstrap P value: we set $\alpha = p^*_1$. The fact that $Q(\alpha,\theta)$ is a rather smooth function of θ suggests that it may not be necessary to compute its value for more than a few different values of θ, and then rely on interpolation.

The discrepancy-corrected bootstrap is computed by the following algorithm. The bootstrap DGP μ^*, characterised by the estimate $\hat\theta$, is used to generate B bootstrap statistics τ^*_j, $j = 1,\dots,B$, from which the first-level bootstrap P value p^*_1 is computed as usual. For each j, the parameter θ^*_j that characterises the double bootstrap DGP μ^{**}_j is computed and saved. Then the *same* random numbers as were used to generate τ^*_j are reused r times with r different values of θ, θ_k, $k = 1,\dots,r$, in the neighbourhood of $\hat\theta$, to generate statistics τ^*_{jk} with τ^*_{jk} generated by the DGP with parameter θ_k. The τ^*_{jk} then allow one to estimate the p^*_1 quantile of the distribution of τ for the DGPs characterised by the θ_k, and the τ^*_j that for $\hat\theta$. The next step is to find by interpolation the value of $Q(p^*_1,\theta^*_j)$ for each bootstrap repetition j. The estimate of the RP of the bootstrap test is then the proportion of the τ^*_j less than $Q(p^*_1,\theta^*_j)$. This algorithm is just the direct way of evaluating the bootstrap discrepancy presented in the previous section, applied to the bootstrap DGP μ^*. The estimated RP is the discrepancy-corrected bootstrap P value, p^*_{DCB}. It requires the computation of $(r+1)B+1$ statistics and $B+1$ bootstrap DGPs. In practice, of course, it is desirable to choose as small a value of r as is compatible with reliable inference.

Simulation evidence

In Figure 5 are shown P value discrepancy curves, as defined in Davidson and MacKinnon (1998), for four bootstrap tests, the conventional (parametric) bootstrap, the double bootstrap, the fast double bootstrap, and the discrepancy-corrected bootstrap. In these curves, the bootstrap discrepancy is plotted as a function of the nominal level α for $0 \leq \alpha \leq 1$. Although it is unnecessary for testing purposes to consider the bootstrap discrepancy for levels any greater than around 0.1, displaying the full plot allows us to see to what extent the distribution of the bootstrap P value differs from the uniform distribution U(0,1). All the plots are based on 10,000 replications with 399 bootstrap repetitions in each.

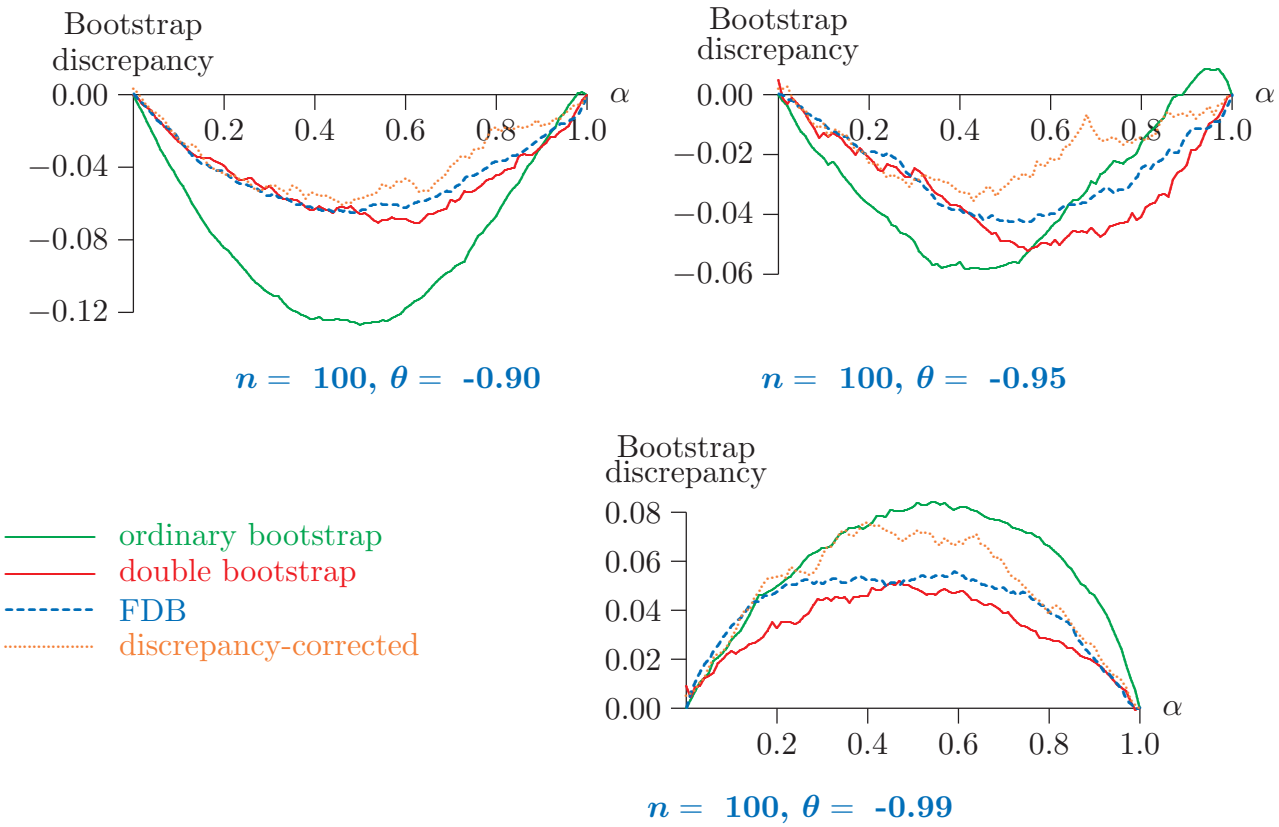

Figure 5: P value discrepancy plots

For the discrepancy-corrected bootstrap, the number r of DGPs used in the simulation was set equal to 4. Two of the values of θ were $\hat{\theta} + 0.02$ and $\hat{\theta} + 0.04$. The third was halfway between $\hat{\theta}$ and -1; the fourth was -1 itself.

In order to emphasise just how small the size distortions are at conventional levels, Table 1 gives the actual numbers for $n = 100$, $\alpha = 0.01$, 0.05, 0.10, and for $\theta = -0.90$, -0.95, -0.99.

Overall, it appears that the corrected bootstrap methods do improve on the ordinary bootstrap. It is striking how similar are the performances of all three of these methods. In particular, the error in the rejection probability (ERP) of the FDB, for which the

θ	-0.90			-0.95			-0.99		
level	1%	5%	10%	1%	5%	10%	1%	5%	10%
ordinary	-0.005	-0.025	-0.047	-0.002	-0.012	-0.021	0.004	0.014	0.028
double	-0.002	-0.013	-0.026	-0.001	-0.004	-0.013	0.005	0.013	0.023
fast double	-0.003	-0.014	-0.025	-0.001	-0.004	-0.010	0.007	0.020	0.034
corrected	0.001	-0.010	-0.025	0.002	-0.005	-0.010	0.006	0.017	0.029

Table 1: Size distortions of bootstrap tests, $n = 100$

theoretical justification is rather weak, given that τ and μ^* are by no means independent, is seldom any greater than that of the double bootstrap, and is often smaller.

7. Possible Extensions

Let $D(\tau, \mu)$ be defined by

$$D(\tau, \mu) = R\Big(Q^*\big(R(\tau, \mu), \mu\big), \mu\Big).$$

Then we can see from (21) that $p^*_{\text{FDB}}(\omega, \mu) = D\big(\tau(\omega, \mu), \mu(\omega, \mu)\big)$. If we can generate $D(\tau, \mu)$ for arbitrary (τ, μ), we can readily generate the independent copies of p^*_{FDB} needed to estimate the RP of the FDB test, and thus obtain a corrected P value for the fast double bootstrap.

In the case in which the space of DGPs is one-dimensional, we can generate $D(\cdot, \mu)$ for a grid of values of μ, and use interpolation for arbitrary μ. For fixed μ, we proceed as follows. For $i = 1, \ldots, N$,

- Generate τ^*_i and μ^*_i as $\tau(\omega_i, \mu)$ and $\mu(\omega_i, \mu)$.
- Generate τ^{**}_i as $\tau(\omega^*_i, \mu^*_i)$. The τ^{**} are IID realisations of the distribution with CDF $R^*(\cdot, \mu)$.
- Sort the pairs (τ^*_i, μ^*_i) in increasing order of the τ^*_i.
- Sort the τ^{**}_i in increasing order.
- Estimate $q^*_i \equiv Q^*\big(R(\tau^*_i, \mu), \mu\big)$ as element i of the sorted τ^{**}. Element i is an estimate of the i/N quantile of the distribution with CDF $R^*(\cdot, \mu)$, that is, of $Q^*(i/N, \mu)$. But, after sorting, τ^*_i estimates the i/N quantile of the distribution with CDF $R(\cdot, \mu)$, and so i/N is an estimate of $R(\tau^*_i, \mu)$.
- Estimate $R\big(Q^*(R(\tau^*_i, \mu), \mu), \mu\big)$ by the proportion of the τ^* that are less than q^*_i.

This gives estimates of $D(\tau, \mu)$ for the given μ and all the realised values τ^*_i. We repeat this for all the μ of our one-dimensional grid.

The next step is to generate the $p^*_{\text{FDB}}(\omega_i, \mu) = D(\tau^*_i, \mu^*_i)$. Since we sorted the μ^*_i along with the τ^*_i, they are still paired. For each of the μ_k, $k = 1, \ldots, K$, of the K points of

the grid, we evaluate $D(\tau_i^*, \mu_k)$ by simple linear interpolation. This is necessary because the realised τ_i^* are different for different μ_k. However, for N large enough, the realisations should be fairly densely spread in the relevant region, and so linear interpolation should be adequate. However, since the experiment for each μ_k is rather costly, we cannot populate the space of the μ at all densely. Thus we prefer to estimate $D(\tau_i^*, \mu_i^*)$ by cubic-spline interpolation based on the $D(\tau_i^*, \mu_k)$. It may be sensible to check that $D(\cdot, \mu)$ is a smooth enough function of μ.

In the experiments done using this approach, $N = 99{,}999$ in the estimation of the grid of values of $D(\tau, \mu)$, but only every 9^{th} realisation was used to compute a realisation of $p^*_{\text{FDB}}(\omega)$, for a total of 11,111 realisations. Computing time was still very long. The results are shown in Figure 6. where the estimated bootstrap discrepancy of the FDB is compared with the discrepancy as estimated by a brute-force simulation experiment. The two estimates are very similar. It is therefore quite possible to envisage a long computation in which the discrepancy of the FDB is bootstrapped, in order to obtain a corrected value, which should suffer from very little size distortion. A simulation experiment to investigate this would unfortunately be extremely costly, at least with today's equipment.

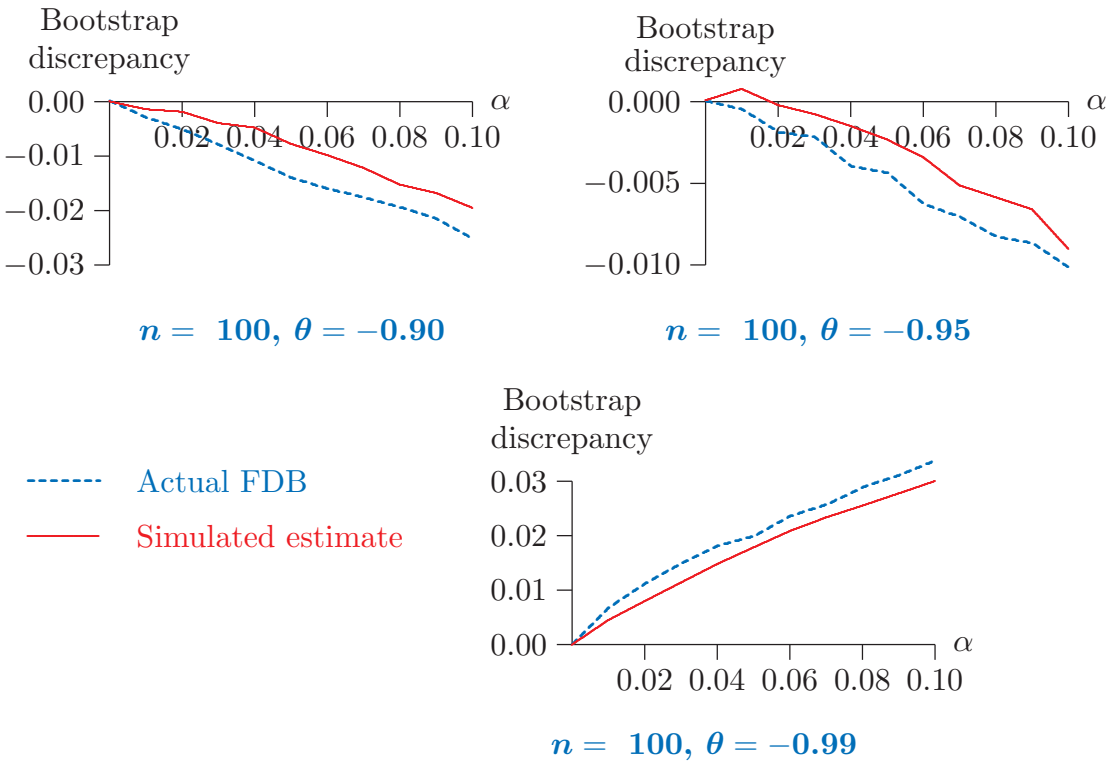

Figure 6: Discrepancy of the FDB

8. Power Properties

In this short section, the power of the (ordinary) bootstrap test is examined. Since its ERP is never very great, the rejection probability under DGPs that do not have a unit

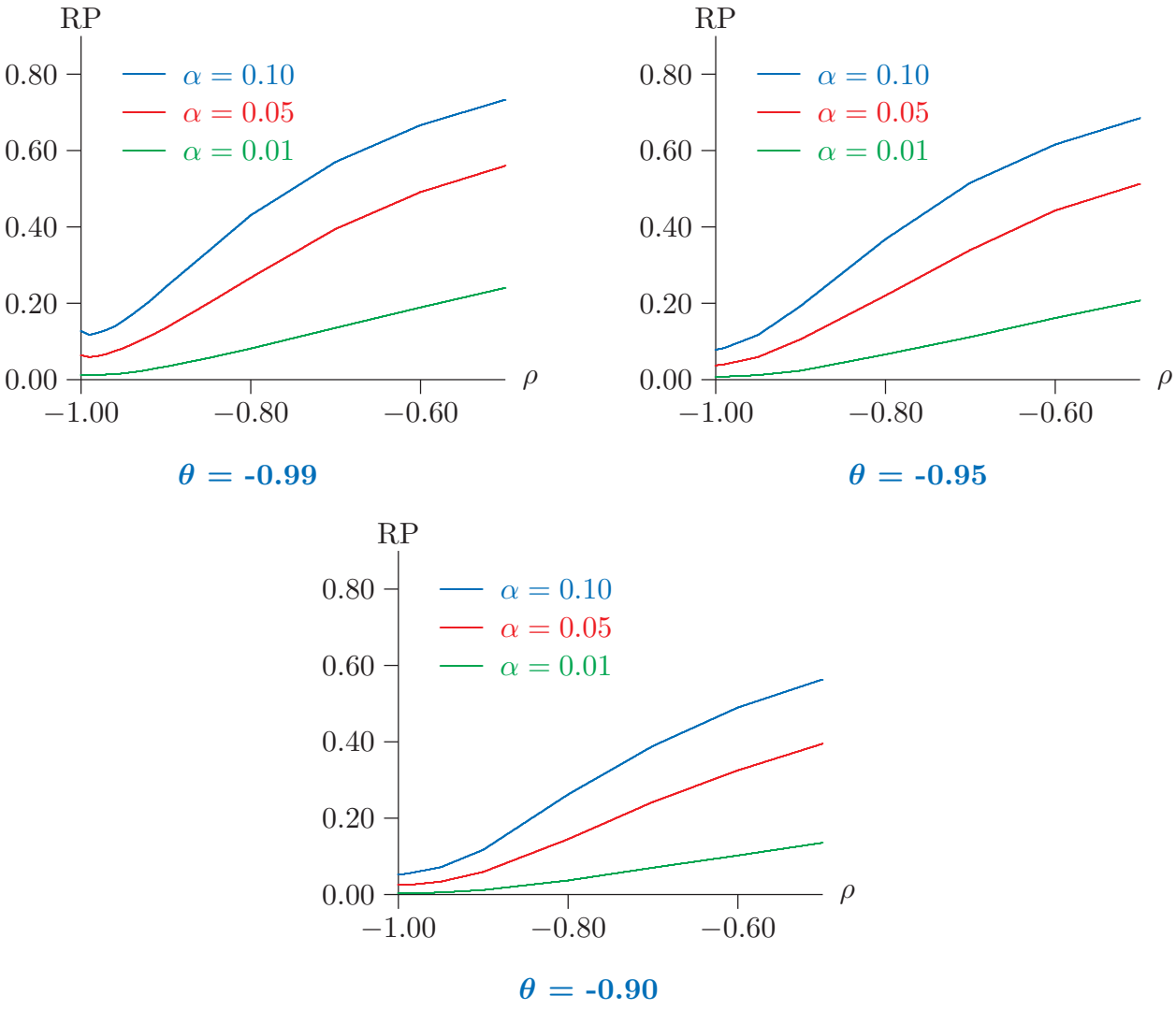

Figure 7: Power of the bootstrap test for $n = 100$

root is a reasonably good measure of the real power of the test. In this specific case, it is in fact possible to consider genuinely size-corrected power, because we have seen that the rejection probability under the null has its supremum when $\theta \to -1$.

In Figure 7, rejection probabilities are plotted for bootstrap tests at nominal levels of 0.01, 0.05, and 0.10, and for sample size $n = 100$, for DGPs that are ARMA(1,1) processes of the form

$$(1 + \rho \mathrm{L})y = (1 + \theta \mathrm{L})\varepsilon, \tag{22}$$

for various values of ρ and θ in the neighbourhood of -1; note that the DGP (13) that satisfies the null hypothesis is just (22) with $\rho = -1$. Whenever $\rho = \theta$, (22) describes only one DGP, whatever the common value of the two parameters may be. This DGP generates a series y that is just white noise. In particular, it is identical to the limit of DGPs that satisfy the null hypothesis with $\rho = -1$ when $\theta \to -1$. It is worth noting that, although a white-noise y might seem very distant from a process with a unit root, it is

in fact the limit of unit-root processes with MA innovations when $\theta \to -1$. The rejection probability under this limiting DGP gives the supremum under the null, which is therefore the size of the bootstrap test for any given nominal level. We see from the Figure that the rejection probability is smaller than the size for ρ closer to -1 than is θ, so that the test is in fact inconsistent against DGPs with such configurations of the two parameters. This is of course not specific to the bootstrap test; it would be true of any ADF test for which it is possible to control Type I error.

Figure 8 shows size-power curves, in which the rejection probability under three alternative DGPs is plotted as a function of the rejection probability under the limiting white-noise DGP. These curves thus plot size-corrected power. The three DGPs each have $\rho = -0.80$, and the three values of θ are -0.99, -0.95, and -0.90. As expected, power falls as θ increases away from -1. The curves show rejection probabilities under the alternatives for all nominal levels, as a way of displaying graphically the difference in the distribution of the bootstrap P value under the null and the alternatives, although, as a practical matter, levels greater than about 0.10 are of no great interest.

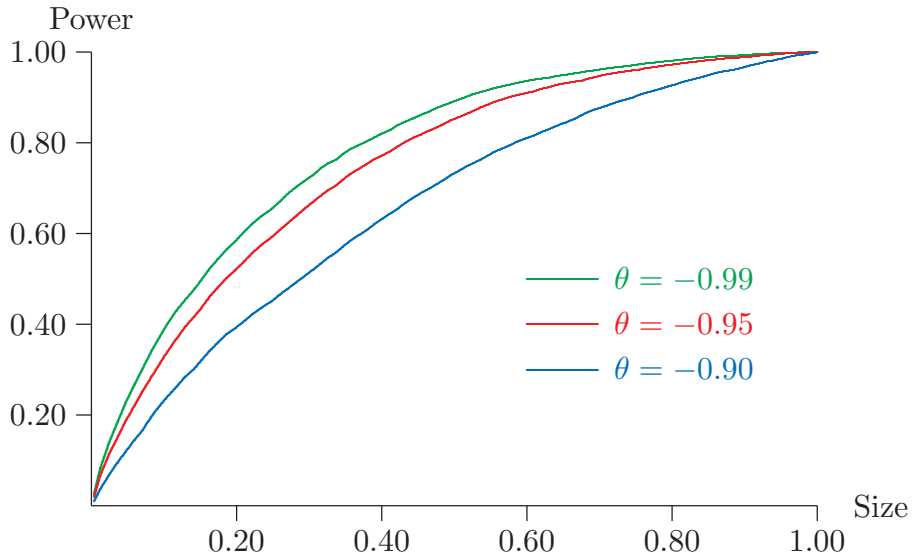

Figure 8: Size-corrected power for ρ = -0.80

9. Concluding Remarks

The focus of this paper is obviously the bootstrap rather than unit-root testing. Quite unconventionally, no use is made of any asymptotic concepts. Asymptotic theory has so far not succeeded in giving a fully satisfactory account of the properties of bootstrap tests in finite samples; indeed the bootstrap often seems to give more reliable inference than asymptotic theory would suggest. Here, although no analytical expressions are given for the finite-sample distributions of the test statistics considered, theory shows that the bootstrap discrepancy depends on the critical-value function $Q(\alpha, \mu)$, which, since the DGP μ in the special case treated here is determined by the scalar parameter θ, can

readily be estimated by simulation combined with interpolation. With estimates of $Q(\alpha, \theta)$ available, we can estimate the distributions of the random variables that determine the bootstrap discrepancy. Comparison of the estimates of the theoretical expression of the bootstrap discrepancy are found to be very close to estimates of the discrepancy obtained by brute-force simulation. It is seen that the most important factor making for bootstrap reliability is the reliability of the estimator(s) that determine the bootstrap DGP. By using a reliable estimator of the MA parameter, one can achieve inference with very little size distortion even when the parameter is close to -1.

The ability to estimate the bootstrap discrepancy leads naturally to the possibility of correcting the bootstrap P value, by bootstrapping the bootstrap discrepancy. The simulations of Section 6 show that the correction provided by the discrepancy-corrected bootstrap is at least as good as that of the much more computationally intensive double bootstrap. Somewhat surprisingly, the still less computationally intensive fast double bootstrap also provides correction similar to that of the other two corrected bootstraps.

The power properties of the bootstrap test, as studied here by simulation, are not at all surprising. The low power for modest sample sizes is an intrinsic feature of unit-root testing; it is just more visible here precisely because the size distortion is so small.

The discrepancy-corrected bootstrap as used in this paper would be much more computationally intensive in cases in which more than one parameter is needed to specify the bootstrap DGP. Nonetheless, it points in a direction that merits a good deal of further study aimed at elucidating the finite-sample behaviour of the bootstrap, and at improving the reliability of bootstrap inference.

References

Beran, R. (1988). "Prepivoting test statistics: a bootstrap view of asymptotic refinements," *Journal of the American Statistical Association*, **83**, 687–697.

Bühlmann, P. (1997). "Sieve bootstrap for time series", *Bernoulli*, **3**, 123–48.

Bühlmann, P. (1998). "Sieve bootstrap for smoothing in nonstationarity time series", *Annals of Statistics*, **26**, 48–83.

Choi, E. and P. Hall (2000). "Bootstrap confidence regions computed from autoregressions of arbitrary order", *Journal of the Royal Statistical Society* series B, **62**, 461–77.

Davidson, R. and J. G. MacKinnon (1998). "Graphical Methods for Investigating the Size and Power of Hypothesis Tests," *The Manchester School*, **66**, 1-26.

Davidson, R. and J. G. MacKinnon (1999). "The Size Distortion of Bootstrap Tests", *Econometric Theory*, **15**, 361–376.

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. (1959). "Efficient estimation of parameters in moving-average models", *Biometrika*, **46**, 306–16.

Galbraith, J. W. and V. Zinde-Walsh (1994). "A simple noniterative estimator for moving-average models", *Biometrika* **81**, 143–55.

Galbraith, J. W. and V. Zinde-Walsh (1997). "On some simple autoregression-based estimation and identification techniques for ARMA models", *Biometrika* **84**, 685–696.

Galbraith, J. W. and V. Zinde-Walsh (1999). "On the distributions of augmented Dickey-Fuller statistics in processes with moving average components", *Journal of Econometrics* **93**, 25–47.

Park, J. Y. (2002). "An invariance principle for sieve bootstrap in time series", *Econometric Theory*, **18**, 469–90.

Park, J. Y. (2003). "Bootstrap unit root tests", *Econometrica*, **71**, 1845–95.

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.

Richard, P. (2007a). "GLS Bias Correction for Low Order ARMA models", Cahiers de recherche from Département d'Economique de la Faculté d'administration à l'Universite de Sherbrooke.

Richard, P. (2007b). "Sieve Bootstrap Unit Root Tests", Cahiers de recherche 07-05, GREDI, Université de Sherbrooke.

Schwert, G. W. (1989). "Testing for unit roots: a Monte Carlo investigation", *Journal of Business and Economic Statistics* **7**, 147–59.