Bid Rigging

An Analysis of Corruption in Auctions[1]

Yvan Lengwiler
University of Basel
Dept. of Economics (WWZ)
Petersgraben 51
CH–4003 Basel
Switzerland
yvan.lengwiler@unibas.ch

Elmar Wolfstetter
Humboldt University at Berlin
Institute of Economic Theory I
Spandauer Str. 1
D–10099 Berlin
Germany
elmar.wolfstetter@rz.hu-berlin.de

April 2005

[1]This article succeeds our earlier working paper entitled "Auctions and Corruption," which circulated in various versions since the year 2000. We wish to thank Dirk Engelmann and Nicolas Sahuguet for their comments, and seminar participants at the Universities of Zürich, Bern, Bergen, Melbourne, Queensland, the Stockholm School of Economics, the University of Korea, the SAET meeting 2001, and the FEEM meeting on "Auctions and Market Design" 2002 for interesting discussions. Financial support was received from the *Deutsche Forschungsgemeinschaft,* SFB Transregio 15, "Governance and Efficiency of Economic Systems."

Abstract

In many auctions, the auctioneer is an agent of the seller. This invites corruption. We propose a model of corruption in which the auctioneer orchestrates bid rigging by inviting a bidder to either lower or raise his bid, whichever is more profitable. We characterize equilibrium bidding in first- and second-price auctions, show how corruption distorts the allocation, and why both the auctioneer and bidders may have a vested interest in maintaining corruption. Bid rigging is initiated by the auctioneer after bids have been submitted in order to minimize illegal contact and to realize the maximum gain from corruption.

JEL classifications: D44.

Keywords: auctions, procurement, corruption, right of first refusal, numerical methods.

1 INTRODUCTION

Corruption is generally defined as the "misuse of a position of trust for dishonest gain." In an auction context, corruption refers to the lack of integrity of the auctioneer. It occurs whenever the auctioneer twists the auction rules in favor of some bidder(s) in exchange for bribes. Corruption may be a simple bilateral affair between one bidder and the auctioneer, but it may also involve collusion among several bidders who jointly strike a deal with the auctioneer.

Corruption is a frequently observed and well documented event in many government procurement auctions. For example, the bidding for the construction of a new metropolitan airport in the Berlin area was recently reopened after investigators found out that *Hochtief AG*, the winner of the auction, was enabled to change its bid after it had illegally acquired the application documents of the rival bidder *IVG*.[1] As another example, in 1996 the authorities of Singapur ruled to exclude *Siemens AG* from all public procurements for a period of five years after they determined that Siemens had bribed Choy Hon Tim, the chief executive of Singapur's public utility corporation *PUB*, in exchange for supplying Siemens with information about rival bids for a major power station construction project.[2]

An interesting early case of corruption in auctions is Goethe's dealing with his publisher Vieweg concerning the publication of his epic poem "Hermann and Dorothea" in the year 1797. Eager to know the true value of his manuscript, Goethe designed a clever scheme. He handed over a sealed note containing his reservation price to his legal Counsel Böttiger. At the same time he asked the publisher Vieweg to make a bid and send it to Böttiger, promising publication rights if and only if the bid is at or above Goethe's reserve price, in which case Vieweg would have to pay Goethe's reserve price.

Obviously, in the absence of corruption, Vieweg should have bid his true valuation. On this ground, Moldovanu and Tietzel (1998) credit Goethe for anticipating the Vickrey auction. However, Goethe's legal Counsel Böttiger was not reliable. Indeed, Böttiger opened Goethe's envelope, and, maliciously informed Vieweg about its content, before he made his bid.[3] Not surprisingly, Vieweg's bid was exactly equal to Goethe's reserve price, and thus Goethe's clever scheme fell prey to corruption.

More recently, the Attorney General of New York, Eliot Spitzer, accused major insurance brokers in the U.S., most prominently Marsh & McLennan, of extensive bid rigging and sued them for fraud and antitrust violations. Apparently, these insurance brokers rigged bids in order to allocate contracts to bidders in exchange for bribes (which they named "contingent commissions"). Allegedly, these bribes

[1] See *Wall Street Journal*, Aug. 19, 1999.

[2] See *Berliner Zeitung*, Feb. 2, 1996.

[3] The letter from Böttiger to Vieweg has been preserved: "Das versiegelte Billet mit dem eingesperrten Goldwolf liegt wirklich in meinem Büreau. Nun sagen Sie also, was Sie geben können und wollen? Ich stelle mich in Ihre Lage, theuerster Vieweg, und empfinde, was ein Zuschauer, der Ihr Freund ist, empfinden kann. Nur eines erlauben Sie mir ... anzufügen: unter 200 Fr[riedrichs]d'or [=1.000 Taler] können Sie nicht bieten." (Jensen, 1984, p. 651).

were in the order of $845m just in the year 2003.[4]

Of course, corruption in an auction cannot occur if the seller is also the auctioneer. Corruption is only an issue if the auctioneer is the agent of the seller. Such delegation occurs if the seller lacks the expertise to run the auction himself, or if the seller is a complex organization like the collective of citizens of a community, a state, or an entire nation. It does not matter whether the auctioneer-agent is a specialized auction house or a government employee. What matters alone is the fact that the auctioneer acts independently on behalf of the seller.

Corruption can also not work in an open-bid auction simply because it lacks secrecy. However, open auctions may not be feasible if the bids are complicated documents, as is the case of auctions for major construction jobs or for the right to host the Olympics. In such auctions sealed bids seems to be the only feasible auction form. The fact that bids are sealed supplies the secrecy needed for corrupt games being played between the auctioneer and one or several bidders.[5]

The literature views corruption in auctions either as a manipulation of the quality assessment in complex bids or as bid rigging. The former was introduced in a seminal paper by Laffont and Tirole (1991), who assume that the auctioneer has some leeway in assessing complex multidimensional bids, and is predisposed to favor a particular bidder. That framework was later adopted by several authors. For example, Celantani and Ganuza (2002) employ it to assess the impact of increased competition on the equilibrium corruption. Paradoxically, they show that corruption may increase if the number of competing bidders is increased. More recently, Burguet and Che (2004) extend that framework. They consider a scoring auction, make the assignment of the auctioneer's favorite agent endogenous, and assume that bribery competition occurs at the same time as contract bidding. Their main result is that corruption may entail inefficiency, and that "… the inefficiency cost of bribery is in the same order of magnitude as the agent's [*i.e. auctioneer's*] manipulation capacity" (Burguet and Che, 2004, p. 61, emphasis added).

A second branch of the literature considers a particular form of bid rigging, in which the auctioneer grants a "right of first refusal" to a favored bidder. This right gives the favored bidder the option to match the highest bid and win the auction. In a first-price auction, the favored bidder thus effectively plays a second price auction, whereas the other bidders pay their bid if they win. Typically, that literature treats the favored bidder status as predetermined. In that case efficiency is destroyed because the favored bidder may not have the highest valuation and yet exercise his right. Burguet and Perry (2003) and Arozamena and Weinschelbaum (2004) analyze this model. The attractive feature of that literature is that it can explain how corruption destroys efficiency. Yet, that feature is lost as soon as one makes the selection of the favored bidder endogenous. For instance, Burguet and Perry also consider a variation of their model in which bidders compete

[4] *The Economist*, "Just how rotten?", Oct. 21, 2004.

[5] Most open auctions are actually hybrids between open and sealed-bid auctions, however, since sealed bids are usually permitted and in fact widely used. In the typical English auction, sealed bids are treated in the same way as in a second-price auction.

for the favored bidder status before the auction by submitting bribes to the auctioneer. This restores efficiency because the strongest bidder offers the highest bribe. Similarly, Koc and Neilson (2005) consider a model in which the right to play a second-price auction is sold for a lump sum bribe before the auction. In that game, only high valuation bidders buy that right, which immediately implies efficiency.

An implausible feature of that approach is that the corrupt auctioneer approaches *all* bidders in order to select the favored one. This entails that the auctioneer exposes himself to an exceedingly high risk of detection and punishment. Every auctioneer who cares about the risk of detection will only propose corruption to the smallest possible number of bidders.

This takes us to the third branch of the literature, to which the present paper belongs. Its key feature is that bid rigging is arranged by the auctioneer after he has observed all the bids. This allows him to approach only a minimum number of bidders, and select the bidder(s) whose collaboration delivers the highest profit.

Lengwiler and Wolfstetter (2000) and Menezes and Monteiro (2003) consider a first-price auction where the auctioneer allows the highest bidder to lower his bid, in exchange for a bribe. They show that this game has a monotone symmetric equilibrium, which implies that efficiency is preserved.[6]

The model by Menezes and Monteiro (2003) has some similarities with our papers, but there are important differences. The most important one concerns the question who the auctioneer should invite to revise his bid. Menezes and Monteiro (2003) assume that the auctioneer either *exclusively* invites the highest bidder to lower his bid or *exclusively* asks the second highest bidder to raise his bid. This is not convincing, because a rational auctioneer chooses the alternative that maximizes his profit, depending upon the submitted bids. Specifically, if the spread between the two highest bids is "large," the auctioneer should propose to the highest bidder to lower his bid, and if that spread is "small" and bid shading is significant, he should propose to the second highest bidder to raise his bid. This is precisely the key feature of our present model.

Finally, we also mention a paper on bid rigging by Compte, Lambert-Mogiliansky, and Verdier (2005). They also assume that the auctioneer allows a bidder to either lower or raise his bid, as we do. However, unlike in our model, they assume that bribes cannot exceed a small upper bound in the order of a small gift and they assume that bribes are offered by all bidders, jointly with bids. As a result, all bidders submit the same maximum bribe and a zero bid, which leads to the interpretation of the role of the corrupt auctioneer as an enforcement device of collusion in the style of the zero bid pooling equilibrium by McAfee and McMillan (1992). The restriction to small bribes may be meaningful in some contexts. However, in many cases bribes are not constrained. To the contrary, corruption often occurs only if bribes are sufficiently large. Moreover, rigging bids *ex post*, after the auctioneer has seen them, minimizes illegal contact and makes the best

[6] Menezes and Monteiro (2003) cover only the case of two bidders; however, they consider two specifications of the bribe: a fixed proportion of the gain from corruption, as in our paper, as well as lump-sum bribes.

use of information. A model of corruption without a small bribes constraint and with *ex post* bid rigging has radically different implications, as we show here.

The present paper analyzes bid rigging in first- and second-price auctions assuming that the auctioneer proposes corruption to the smallest possible coalition of bidders that makes corruption feasible. The selection of the most profitable partners in crime and the restriction to the smallest number of parties involved is made possible by arranging bid rigging after bids have been submitted. In the first-price auction, corruption involves only one bidder, in the second-price auction it typically involves two.

Two kinds of corruption must be distinguished: either one bidder is invited to lower his bid (we call this *type I corruption*) or one bidder is allowed to match the highest bid, as in a right of first refusal arrangement (we call this *type II corruption*). Before the auctioneer proposes corruption he compares the bids and decides which of the two kinds of corruption is the most profitable. In general that choice depends upon the bids, and the degree of bid shading.

Our main results are as follows: if the auction is second-price, corruption exclusively takes the form of a reduction of the second highest bid (type I corruption), which lowers the price paid by the highest bidder. In that case, corruption affects the distribution, but not the allocation. If the auction is first-price, both types of corruption are involved. In the event when the spread between the two highest bids is "large," type I corruption takes place, and if that spread is "small," type II corruption is chosen. In the latter event corruption destroys efficiency. Altogether, both the auctioneer and bidders benefit from corruption. This may explain why fighting corruption is intrinsically difficult.

The second-price auction can be solved explicitly. This is not true for the first-price auction. There, the interplay between the two types of corruption gives rise to a delayed differential equation problem that has no closed form solution. Therefore, we apply numerical methods to approximate the equilibrium and to study its properties.

2 THE MODEL

There is one seller of a single good who faces $n \geqslant 2$ risk neutral potential buyers. The analysis is carried out in the standard symmetric, independent, private values framework. The seller fully delegates to an auctioneer to run either a sealed-bid first- or second-price auction. Unlike in the standard model, the auctioneer is corrupt, and this fact is common knowledge among bidders. In the following, bids are denoted by $b_1, \dots, b_n$. W.l.o.g. we order bidders in such a way that $b_i \geqslant b_{i+1}$.

In our model, corruption means that the auctioneer allows one bidder to revise his bid in exchange for a bribe. In the sealed-bid first-price auction, the corruption game is as follows: after bids have been submitted, the auctioneer can invite the highest bidder to lower his bid to b_2 (this is type I corruption). The surplus from corruption that the auctioneer and the winning bidder can share is the reduction of the price payed to the seller, $b_1 - b_2$. Alternatively, the auctioneer can propose the second highest bidder to raise his bid from b_2 to b_1 (type II corruption). In this case, the surplus from corruption is the difference between the valuation of the second highest bidder and the highest bid, $v_2 - b_1$. In both cases the bidder receives a share α and the auctioneer a share $(1-\alpha)$ of the surplus. If the proposal for corruption is rejected, the auctioneer sticks to the official rules of the game, because he tries corruption once and only once.

In the sealed-bid second-price auction, the corruption game is slightly more involved: the auctioneer either permits the second highest bidder to remove his bid (type I), which lowers the price paid by the highest bidder from b_2 to b_3, or permits the second highest bidder to raise his bid from b_2 to b_1 (type II). In these arrangements, three parties are involved in corruption, namely the auctioneer and the two highest bidders. We assume equal sharing among the two involved bidders, so each of them receives a share $\alpha \leqslant 1/2$, and the auctioneer receives $1 - 2\alpha$. Again, if the proposal for corruption is rejected by one of the bidders, the auctioneer sticks to the original rules, because he tries corruption once and only once.

Can there be also type III or IV etc., that is, corruption that involves a deal with the third or fourth highest bidder? The answer is no if the equilibrium bid function is monotone increasing. Deals with either the third highest or lower bidders are then less profitable than a deal with the second highest bidder, because the valuation of these bidders is lower, while the price they have to pay cannot be smaller than b_1.

We assume that the winning bid is published after the result is communicated to the seller. This publication requirement is common practice. For example, U. S. Federal Law mandates the publication of the winning bid in all government procurements. This assumption is important because it makes it impossible for the auctioneer to propose a deal to some arbitrary bidder and sidestep the highest bidder. We also assume that the auctioneer can document bids to the bidders, and cannot forge new bids. He needs the cooperation of a bidder to revise a bid in a corrupt deal.

Valuations $v_1, \dots, v_n$ are independent draws from the continuously differentiable c.d.f. F with support $[0, 1]$ and p.d.f. $f(v) := F'(v)$. From the perspective of one bidder, the valuations of all rival bidders is a random sample of size $n - 1$. We denote the highest and second highest of these $n - 1$ valuations by the order statistics Y_1 and Y_2. The probability distribution function of Y_1 is $G(x) := \Pr\{Y_1 \leqslant x\} = F(x)^{n-1}$. The joint density function of the order statistics Y_2 and Y_1 is

$$f_{Y_2 Y_1}(z, y) = (n-1)(n-2)F(z)^{n-3} f(z) f(y), \quad \text{for } z \leqslant y, \tag{1}$$

and 0 otherwise (see David, 1970, p. 10).

A *bidding strategy* is a map $\beta^i \colon v_i \mapsto b_i$. An *equilibrium* is a profile of strategies $(\beta^1, \dots, \beta^n)$ such that β^i is a best reply for i given the strategies of all other bidders. An equilibrium is *symmetric* if all bidders use the same strategy, $\beta^1 = \cdots = \beta^n$.

We denote the symmetric equilibrium bidding strategy with β. We call the corresponding auctions in which corruption is not part of the game the *standard first- and second-price auctions*, respectively, and denote the respective symmetric equilibrium bid functions with B_1 and B_2.

In principle, both types of corruption are conceivable in the second-price auction. The auctioneer could invite the second highest bidder to match the highest bid (type II corruption), or he could invite the second highest bidder to withdraw his bid (type I corruption), in exchange for a bribe that is payed for by the highest bidder. However, as we show in this section, assuming that type II corruption does not occur, we actually find an equilibrium in which this assumption confirms because type II corruption is never profitable.

Assume, as a working hypothesis, that only type I corruption is contemplated. Of course, this type of corruption requires the collaboration of at least two bidders because the winning bidder and the auctioneer alone cannot change the price. Therefore, the smallest coalition that makes type I corruption profitable involves the highest and the second highest bidder. If corruption succeeds, the winning bidder pays only the third highest bid. The gain from corruption is then shared by the auctioneer, the highest bidder, and the second highest bidder.

The case of two bidders is somewhat special, because in that case the withdrawal of the second highest bid reduces the price all the way down to the reserve price, which we normalize to zero. Therefore, in the following, we distinguish between the cases $n = 2$ and $n \geqslant 3$.

To solve the second-price auction with corruption we proceed as follows. Assume the equilibrium is symmetric and the equilibrium strategy β is strict monotone increasing (this will be confirmed later on).

Consider one bidder and suppose all other bidders play the strategy β. Then that bidder need only consider bids from the range $[\beta(0), \beta(1)]$, since bidding outside this range is either unnecessarily high or too low. Therefore, all relevant deviating bids can be generated by inserting $x \in [0, 1]$ into the equilibrium strategy, β, i.e. by bidding as if the valuation were $x \in [0, 1]$ rather than the true valuation v. Let $U(v, x)$ denote the deviating bidder's payoff if he makes the bid $\beta(x)$, and all rival bidders play the strategy β. Then, for $n \geqslant 3$, $U(v, x)$ can be written as[7]

$$\begin{aligned} U(v,x) = &\int_0^x (v - \beta(y))dG(y) \\ &+ \alpha \int_0^x \int_0^y (\beta(y) - \beta(z)) f_{Y_2 Y_1}(z, y) dz dy \\ &+ \alpha \int_x^1 \int_0^x (\beta(x) - \beta(z)) f_{Y_2 Y_1}(z, y) dz dy. \end{aligned} \tag{2}$$

We characterize the equilibrium bid function β using the equilibrium requirement $v = \arg\max_x U(v, x)$, which leads to the first-order condition,

$$\begin{aligned} 0 = &(v - \beta(v))G'(v) \\ &+ \alpha \beta'(v) \int_v^1 \int_0^v f_{Y_2 Y_1}(z, y) dz dy. \end{aligned} \tag{3}$$

[7]Recall that the winning bidder receives a share α of the gain from corruption whether he wins through type I or type II corruption.

Using the definition of f_{Y_2,Y_1}, equation (1), and the fact that $(n-1)F(v)^{n-2} = G'(v)/f(v)$, this first-order condition simplifies to

$$(\beta(v) - v)G'(v) = \alpha\beta'(v)\frac{G'(v)}{f(v)}(1 - F(v)). \tag{4}$$

For $n = 2$ the payoff function is

$$\begin{aligned} U(v,x) = &\int_0^x (v - \beta(y))dF(y) + \alpha\int_0^x \beta(y)dF(y) \\ &+ \alpha(1 - F(x))\beta(x). \end{aligned} \tag{5}$$

This yields the first-order equilibrium requirement

$$(\beta(v) - v)f(v) = \alpha\beta'(v)(1 - F(v)). \tag{6}$$

This is a special case of (4), because $G' = f$ if $n = 2$. Thus, (4) applies to all $n \geqslant 2$, even though the payoff functions differ for $n = 2$ and for $n > 2$.

Proposition 1 (Restricted Second-Price Auction) *The symmetric equilibrium bid function, β, and bidders' equilibrium payoff function, u, are*

$$\beta(v) := \begin{cases} v + \int_v^1 \frac{K(y)}{K(v)}dy, & \text{if } \alpha > 0 \text{ and } v < 1, \\ B_2(v) = v, & \text{if } \alpha = 0 \text{ or } v = 1. \end{cases} \tag{7}$$

$$K(v) := (1 - F(v))^{1/\alpha} \tag{8}$$

$$u(v) := U(v,v) = \int_0^v G(y)dy + u(0), \tag{9}$$

$$u(0) := \begin{cases} \alpha\beta(0) = \alpha\int_0^1 K(y)dy, & \text{if } n = 2 \text{ and } \alpha > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

β is independent of n, and $\beta(v) > v, \forall v \in [0,1)$.

Proof: If $\alpha = 0$, truthful bidding follows immediately from (4). Now suppose $\alpha > 0$. Because $(1 - F(v))' = -f(v)$, (4) can be further simplified to

$$(v - \beta(v))(1 - F(v))' = \alpha\beta'(v)(1 - F(v)). \tag{11}$$

Substitute F by using the definition $K(v) := (1 - F(v))^{\frac{1}{\alpha}}$. Then, after a bit of rearranging, the differential equation (11) can be rewritten in the form

$$(\beta(v)K(v))' = vK'(v). \tag{12}$$

Since $K(0) = 1$ one gets

$$\beta(v)K(v) - \beta(0) = \int_0^v yK'(y)dy. \tag{13}$$

Using integration by parts, and because $K(1) = 0$, this entails

$$\beta(0) = \int_0^1 K(y)dy, \tag{14}$$

and the asserted bid function for all $v \in [0, 1)$ and its "overbidding" property follow after some rearranging. Finally, using L'Hospital's rule, one finds (assuming that $f(v) > 0$ everywhere)

$$\lim_{v \to 1} (\beta(v) - v) = \lim_{v \to 1} \frac{\int_v^1 K(y)dy}{K(v)} = \lim_{v \to 1} \frac{-K(v)}{K'(v)} = \lim_{v \to 1} \frac{\alpha K(v)^\alpha}{f(v)} = 0, \tag{15}$$

which proves the assertion for $v = 1$.

To compute bidders' equilibrium payoffs, note that if $n = 2$, a bidder with valuation $v = 0$ earns a positive payoff, because he is bribed to lower his bid from $\beta(0) > 0$ to 0; whereas if $n \geqslant 3$, such a bidder is not part of a corruption scheme with probability 1. Therefore,

$$u(v) := U(v, v) = \int_0^v G(y) + u_2(0), \tag{16}$$

$$u(0) = \begin{cases} \alpha\beta(0) = \alpha \int_0^1 K(y)dy, & \text{if } n = 2, \\ 0, & \text{if } n \geqslant 3. \end{cases} \tag{17}$$

□

Remark 1 *If $n = 2$, the equilibrium problem is equivalent to that of an auction where both, winner and loser, receive an equal share α of the price. Such auctions were analyzed in the partnership dissolution literature (see Cramton, Gibbons, and Klemperer (1987) and especially Engelbrecht-Wiggans (1994)).*[8]

Remark 2 *As is well known, the second-price auction has also asymmetric equilibria. For instance, it is an equilibrium to have one bidder bid an amount that exceeds all possible valuations, and all others to bid zero. This equilibrium is actually corruption-proof, excluding type I and type II corruption. However, this and similar asymmetric equilibria fail standard equilibrium refinements.*

Proposition 2 (Second-Price Auction) *The equilibrium of the Restricted Game in Proposition 1 is also a solution of the Full Game, in which the auctioneer is free to propose type I or type II corruption.*

Proof: As we showed above, the equilibrium of the restricted second-price auction exhibits strict overbidding, i.e., $\beta(v) > v$, for all $v \in [0, 1)$. It follows that there is no room for type II corruption. Because if the auctioneer would propose type II corruption to the second highest bidder, that bidder would have to pay more than his valuation, which he would reject. □

[8] If $n > 2$, these equilibrium problems differ; nevertheless they have the same equilibrium solution for the same bidders' share (although equilibrium payoffs differ), as we show in Lengwiler and Wolfstetter (2005b).

Corollary 1 *In the second-price auction corruption does not destroy efficiency. If $n > 2$ or $\alpha = 0$, corruption benefits only the auctioneer at the expense of the seller. If $n = 2$ and $\alpha > 0$, bidders also benefit from corruption, not only the auctioneer.*

Proof: The proof follows immediately from the revenue equivalence theorem, combined with the fact that $u(0) > 0$ if and only if $n = 2$ and $\alpha > 0$, and $u(0) = 0$ otherwise. □

4 FIRST-PRICE AUCTIONS: TWO IMPOSSIBILITY RESULTS

We now turn to the first-price auction. Again we first consider the restricted game, in which only one type of corruption is contemplated by the auctioneer. We compute its equilibrium and then ask whether it is also an equilibrium of the full game in which both types of corruption are possible.

We start with the restricted game in which only type I corruption is contemplated by the auctioneer. To solve that game, we assume as a working hypothesis that the equilibrium is symmetric and strict monotone increasing, and then confirm this hypothesis.

Let $U(v, x)$ denote a bidder's expected payoff provided 1) his true valuation is v, 2) he bids $\beta(x)$, and 3) all rival bidders play the strategy β. This expected payoff is given by

$$U(v, x) = (v - \beta(x))G(x) + \alpha \int_0^x (\beta(x) - \beta(y))dG(y). \tag{18}$$

The equilibrium requirement $v = \arg\max_x U(v, x)$ takes the form

$$0 = (v - \beta(v))G'(v) - \beta'(v)(1 - \alpha)G(v). \tag{19}$$

Proposition 3 (Restricted First-Price Auction I) *Consider the restricted first-price auction game in which only type I corruption is permitted.*[9] *The equilibrium bid function is*[10]

$$\beta(v) = \begin{cases} v - \int_0^v \frac{H(y)}{H(v)}dy = E_{Y_1 \sim H}[Y_1 \mid Y_1 < v], & \text{if } \alpha < 1, \\ B_2(v), & \text{if } \alpha = 1, \end{cases} \tag{20}$$

$$H(y) := G(y)^{\frac{1}{1-\alpha}}. \tag{21}$$

It is equivalent to the solution of a first-price auction without corruption in which valuations are drawn from the fictitious distribution function H, and for $\alpha < 1$ one has $\beta(v) > B_1(v)$ for all $v \in (0, 1]$.

[9] Menezes and Monteiro (2003, Section 4) also solve this game for $n = 2$.

[10] $E_{Y_1 \sim H}[Y_1 \mid Y_1 < x]$ denotes the conditional expected value of the order statistic Y_1, assuming Y_1 is drawn from the c.d.f. H.

PROOF: Substitute $G(v)$ in (19), using the definition of H, and assume $\alpha < 1$. After a bit of rearranging, (19) can be rewritten in the form

$$\forall v \qquad vH'(v) = (\beta(v)H(v))'. \tag{22}$$

Integrate (22), use integration by parts, and one has (20). If $\alpha = 1$, the payoff function (18) is the same as in the second-price auction without corruption, which proves $\beta = B_2$ in this case.

The asserted equivalency with the as-if equilibrium of the first-price auction without corruption is immediately clear, and so is the asserted $\beta = E_{Y_1 \sim H}[Y_1 \mid Y_1 < v]$.

The necessary condition is also sufficient if 1) the derived strategy β is strict monotone increasing (which was assumed when we wrote the necessary condition) and 2) if each stationary point is indeed a global best reply. The first requirement follows immediately, and the second is implied by the fact that $U(v, x)$ is increasing in x if and only if $x < v$ and decreasing if and only if $x > v$, as we show in Appendix A. □

Compared to the equilibrium of the first-price auction without corruption, the prospect of a higher gain makes all bidders bid more aggressively. Formally, this follows from the fact that the distribution function H first-order stochastically dominates G for all $\alpha < 1$. Therefore, by the first-order stochastic dominance theorem, for all v: $E_{Y_1 \sim H}[Y_1 \mid Y_1 < v] > E_{Y_1 \sim G}[Y_1 \mid Y_1 < v]$.

REMARK 3 *In that equilibrium the winner ends up paying a convex combination of the two highest bids (if $\alpha < 1$), and losers get nothing. Auctions with such pricing rules were analyzed by Güth (1995) and Riley (1989). Therefore, our proof of Proposition 3 is also a particularly simple solution of that auction problem without corruption.*

PROPOSITION 4 (IMPOSSIBILITY RESULT I) *The equilibrium of the Restricted Game I is not also an equilibrium of the full game in which the auctioneer is free to propose either type I or type II corruption.*

PROOF: The equilibrium stated in Proposition 3 is a strict monotone increasing and continuous function. Moreover, it exhibits bid shading for all $v > 0$. Therefore, there are valuations for which the two highest bids are arbitrarily close together. Whenever this occurs, type II corruption is more profitable than type I corruption. □

Here we digress briefly and assess the implications of leniency rules in U.S. corporate law for the stability of corruption. Leniency rules were designed to fight collusion and corruption, by rewarding the "whistle blower." According to these rules, leniency is only granted to bidders, but not to the auctioneer, because the party who initiates the illegal activity is not eligible.[11] Moreover, the original highest bid has to be paid, because the damaged seller must be compensated. This

[11] The leniency policy of the U.S. Department of Justice is summarized at `www.usdoj.gov/atr/public/guidelines/lencorp.htm`.

implies that in the event of type I corruption, the bidder(s) involved cannot gain from "blowing the whistle." As a result, these leniency rules cannot deter type I corruption. In the event of type II corruption, however, the seller is not directly hurt, because the winner actually pays the highest bid. Therefore, the winning bidder can only gain by "blowing the whistle": he need not compensate the seller for damages, avoids detection and punishment, and in addition may reclaim some of the money paid to the corrupt auctioneer. Therefore,

REMARK 4 *Leniency rules may prevent type II corruption, but not type I corruption, and thus justify the equilibrium proposed in Proposition 3.*

Having shown that bid rigging is not restricted to type I corruption (unless leniency rules are effective that prevent type II corruption), next we consider the other restricted first-price auction in which the auctioneer can only propose type II corruption, a case also considered by Menezes and Monteiro (2003, Section 6.3).

Suppose the auctioneer always proposes type II corruption to the second highest bidder. Then, a bidder wins if and only if he submits the second highest bid, and in that event pays the highest bid to the seller plus a share of the gain to the auctioneer. It follows immediately that bidding more than one's valuation is dominated. Every best reply must exhibit bid shading. This property is essential for characterizing the equilibrium.

PROPOSITION 5 (RESTRICTED FIRST-PRICE AUCTION II) *Consider the restricted first-price auction in which only type II corruption is permitted. That game has the following unique pooling equilibrium,*

$$\beta(v) = 0, \quad \forall v. \tag{23}$$

It has no symmetric strict monotone increasing equilibrium.

PROOF: Bidding zero is obviously an equilibrium, because by unilaterally bidding higher one can only lose the auction. There is obviously no other symmetric pooling equilibrium that involves a positive bid.

Next we show that there is no symmetric monotone equilibrium. The proof is by contradiction. Suppose the Restricted Game II has a symmetric monotone equilibrium $\beta(v)$. Consider a bidder who has valuation v and makes a deviating bid $\beta(x)$ with x close to v.[12] That bidder wins the auction if and only if

$$\beta(Y_2) < \beta(x) < \beta(Y_1) < x. \tag{24}$$

Since β is monotone, the inverse $\beta^{-1} : [0, \beta(1)] \to [0, 1]$ exists. Extend the domain from $[0, \beta(1)]$ to $[0, 1]$ by defining the extended inverse: $c(x) := \beta^{-1}(x)$ for all $x \in [0, \beta(1))]$ and $c(x) = 1$ for all $x \in [\beta(1), 1]$. Then, we can apply this generalized inverse to (24) and obtain the equivalent condition,

$$Y_2 < x < Y_1 < c(x) := \beta^{-1} (\min\{x, \beta(1)\}) . \tag{25}$$

[12] A large deviation, such that $\beta(x) > x$, is obviously pointless, because a bidder who deviates to such an extent would refuse an offer to participate in type II corruption, since, if he accepted, he would have to pay more than his valuation.

Therefore, that bidder's payoff function is

$$
\begin{aligned}
U(v,x) &:= \int_x^{c(x)} \int_0^x (v-\beta(y)) f_{Y_2Y_1}(z,y)\,dz\,dy \\
&= (n-1)F(x)^{n-2} \int_x^{c(x)} (v-\beta(y))\,dF(y).
\end{aligned}
\tag{26}
$$

For β to be a symmetric equilibrium, one must have $v = \arg\max_x U(v,x), \forall v$. Therefore, the following first-order equilibrium conditions must hold,

$$
\begin{aligned}
&\text{if } v < \beta(1), \quad (n-2)\int_v^{\beta^{-1}(v)} (v-\beta(y))\,dF(y) - (v-\beta(v))F(v) = 0,\\
&\text{if } v \geqslant \beta(1), \quad (n-2)\int_v^{1} (v-\beta(y))\,dF(y) - (v-\beta(v))F(v) = 0.
\end{aligned}
\tag{27}
$$

Evidently, truthful bidding is a strict monotone increasing solution of this differential equation. Moreover, by (27), every strict monotone solution must have the property $\beta(1) = 1$. However, bidding truthfully yields a zero payoff. By bidding less that bidder could make a positive payoff with positive probability. Thus, $\beta(1) = 1$ is not a best reply, and therefore there is no symmetric strict monotone equilibrium. □

COROLLARY 2 *Proposition 5 applies also the the restricted second-price auction game in which only type II corruption is permitted.*

PROOF: In both auctions, the second highest bidder is invited to match the highest bid. The two highest bids after corruption has taken place are therefore equal to each other. Hence, the payoff functions in the two games are identical. □

PROPOSITION 6 (IMPOSSIBILITY RESULT II) *The symmetric equilibrium of the Restricted Game II is not also an equilibrium of the full game in which the auctioneer is free to propose either type I or type II corruption.*

PROOF: Suppose all bidders play the equilibrium strategy of the Restricted Game II, $\beta(v) = 0, \forall v$, in the full game. We show how the auctioneer continues to play if he observes zero bids by all players, and we then construct a profitable deviating bid that induces the auctioneer to propose type I corruption.

If the auctioneer observes zero bids by all players he selects one bidder at random, proposes type II corruption to him, requesting a bribe t that maximizes the gain from corruption (given his prior beliefs about bidders' valuations), If that proposal is rejected, he picks a winner at random and makes him pay zero.[13] If

[13] Note, if the auctioneer asks one randomly selected bidder to raise his bid to t, that bidder accepts with probability $1 - F(t)$. Therefore, the optimal t that maximizes the gain from corruption is the maximizer of $t(1 - F(t))$. It is unique if the hazard rate is monotone increasing; if it is not unique, one must specify which maximizer (say the largest) is selected.

the auctioneer observes a deviating bid $b' \neq b$, he either proposes type II corruption to a non-deviating bidder, requesting a bribe equal to t (just as he does on the equilibrium path), or proposes type I corruption to the deviating bidder, whichever is more profitable.

Now consider a bidder who has a sufficiently high valuation, $v > t$, and who plays the deviating strategy $b' = t/(1-\alpha)$, while all other bidders play $b = 0$. Then, that bidder will be proposed type I corruption (which is independent of the auctioneer's beliefs),[14] win for sure, pay the bribe t to the auctioneer, and thus earn the profit $v - t > 0$, with certainty. This is obviously better than sticking to the pooling strategy $b = 0$, which earns the profit $v - t$ only with probability less than one. Therefore, $\beta(v) = 0$ is not an equilibrium of the full game. □

We conclude that the analysis of the first-price auction must incorporate both types of corruption, to which we now turn.

5 FIRST-PRICE AUCTIONS WITH BOTH TYPES OF CORRUPTION

We now analyze the first-price auction if both type I and type II corruption may occur. We already showed that this game has no pooling equilibrium (see Proposition 5 and 6), and now focus on separating equilibria. The gain from type II corruption is equal to $v_2 - b_1$, and the gain from type I corruption is $b_1 - b_2$. Therefore, the auctioneer proposes type II corruption if and only if there is bid shading, and the bid spread is sufficiently small, i.e. $b_1 - b_2 \leqslant v_2 - b_1$. However, the auctioneer only observes bids, not valuations. Therefore, in order to assess the gain from type II corruption, he has to infer the valuation that underlies the observed bid, b_2. This introduces a signalling aspect into the bidding problem.

In a separating equilibrium the bid function is strict monotone increasing. Therefore, the auctioneer can draw an exact inference from an observed equilibrium bid b from the image set of β to the underlying valuation, using the inference rule: $\beta^{-1}(b) = v$. If he observes an off-equilibrium bid $b > \beta(1)$, we assume that the auctioneer infers $v = 1$, and if he observes any other off-equilibrium bid he infers $v = 0$. Therefore, the auctioneer proposes type II corruption if and only if $b_1 - b_2 \leqslant x_2 - b_1$, where $x_2 := \beta^{-1}(b_2)$; and, that proposal is accepted if and only if $v_2 - b_1 - (1-\alpha)(x_2 - b_1) > 0$, which is assured, unless the second highest bidder has bid considerably higher that the equilibrium bid.

After these preliminaries, we now analyze the separating equilibrium in detail. For this purpose, suppose a bidder has submitted a bid $\beta(x)$ with x not necessarily equal to that bidder's valuation v, and that bid happens to be the second highest. The valuation of the highest rival bidder is denoted by y, and the corresponding highest rival bid by $\beta(y)$. The bid function β is assumed to be strict monotone increasing. The auctioneer *proposes* type II corruption to the second highest bidder,

[14] Note that the signaling of a bidder's valuation is only relevant for the assessment of the gain from type II corruption. The assessment of the gain from type I corruption is independent of the auctioneer's beliefs.

with bid $\beta(x)$ if and only if $x - \beta(y) \geqslant \beta(y) - \beta(x)$. Define

$$\overline{\psi}(x) := \begin{cases} \beta^{-1}\left(\frac{x+\beta(x)}{2}\right) =: \psi(x), & \text{if } (v+\beta(v))/2 < 1, \\ 1, & \text{if } (v+\beta(v))/2 \geqslant 1. \end{cases} \tag{28}$$

The auctioneer proposes type II corruption if and only if

$$y \leqslant \overline{\psi}(x). \tag{29}$$

In exchange, he demands a transfer $(1-\alpha)(x-\beta(y))$. In turn, that highest losing bidder accepts this deal if and only if $v - \beta(y) - (1-\alpha)(x-\beta(y)) \geqslant 0$, which is equivalent to

$$y \leqslant \beta^{-1}\left(\frac{v-(1-\alpha)x}{\alpha}\right) =: \tilde{\psi}(v,x). \tag{30}$$

Type II corruption takes place if both constraints, (29) and (30), are satisfied. Therefore, the second-highest bidder with valuation v who bids $\beta(x)$ will be part of type II corruption if and only if the highest valuation of rival bidders, y, satisfies the condition

$$x \leqslant y \leqslant \min\{\overline{\psi}(x), \tilde{\psi}(v,x)\} =: \psi^*(v,x). \tag{31}$$

An immediate implication of (29) and (30) is that whenever the auctioneer proposes type II corruption, the bidder is sure to accept, unless that bidder exaggerates his valuation considerably, since

$$(29) \Rightarrow (30) \quad \text{if} \quad x \leqslant v + \frac{\alpha}{1-\alpha}(v - \beta(y)). \tag{32}$$

Similarly, a bidder is part of type I corruption if he is the highest bidder ($x > y$), and the second highest bidder is not offered type II corruption, which requires that y is not too close to x, in the sense that

$$y < \phi(x), \quad \text{where} \quad \phi(x) + \beta(\phi(x)) \equiv 2\beta(x). \tag{33}$$

The bidder who is offered type I corruption always accepts because he can only gain.

Notice that $\psi(\phi(x)) = x$. Figure 1 depicts the different types of corruption as a function of the two highest bids, $\beta(x)$ and $\beta(y)$.

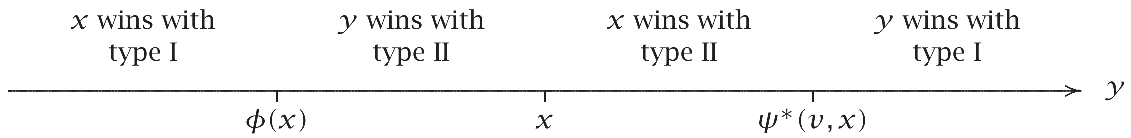

Figure 1: Regions of type I and type II corruption.

We now characterize the equilibrium bid function, separately for the case of $n \geqslant 3$, and for the somewhat special two bidder case.

The payoff of a bidder with valuation v who bids as if his valuation were x, while all others bid the symmetric, strict monotone increasing equilibrium strategy β, is, for $n \geqslant 3$,

$$U(v,x) := \int_0^{\phi(x)} (v - \beta(y) - (1-\alpha)\,(\beta(x) - \beta(y)))\, dG(y) + \int_x^{\psi^*(v,x)} \int_0^x (v - \beta(y) - (1-\alpha)\,(x - \beta(y)))\, f_{Y_2Y_1}(z,y) dz dy. \quad (34)$$

Using the definition of the joint density $f_{Y_2Y_1}$ (see (1)), one can simplify the second line of that payoff function, using the fact that

$$\int_0^x f_{Y_2Y_1}(z,y)dz = (n-1)f(y)F(x)^{n-2} = G'(x)\frac{f(y)}{f(x)}. \quad (35)$$

The strategy β is a symmetric equilibrium if $v = \arg\max_x U(v,x)$ for all v. Using the first-order condition of this requirement (keeping in mind that $\psi^*(v,x) = \overline{\psi}(x)$ for x in a neighborhood of v, by (32)), one obtains

$$\begin{aligned} 0 = {} & \phi'(v)(v - (1-\alpha)\beta(v) - \alpha\beta(\phi(v)))G'(\phi(v)) \\ & - (1-\alpha)\beta'(v)G(\phi(v)) \\ & + \alpha(v - \beta(v))\frac{G'(v)}{f(v)}\left[\frac{\overline{\psi}'(v)f(\overline{\psi}(v))}{2} - f(v)\right] \\ & - (1-\alpha)\frac{G'(v)}{f(v)}(F(\overline{\psi}(v)) - F(v)) \\ & + \alpha\int_v^{\overline{\psi}(v)} (v - \beta(y))f_{Y_2Y_1}(v,y)dy, \end{aligned} \quad (36)$$

where

$$\overline{\psi}'(v) := \begin{cases} \psi'(v) & \text{if } (v + \beta(v))/2 < 1, \\ 0 & \text{if } (v + \beta(v))/2 \geqslant 1. \end{cases}$$

We now turn to the case of $n = 2$, which is slightly special. When there are three or more bidders, a bidder is proposed type II corruption if three conditions are met: first, his bid must be lower than the highest rival bid, but second, not too much lower, and third, it must be higher than the second highest rival bid. When there are only two bidders, the third requirement is meaningless. Therefore, the payoff function and the first-order condition simplify to

$$
\begin{aligned}
U(v,x) = &\int_0^{\phi(x)} (v - \beta(y) - (1-\alpha)(\beta(x) - \beta(y)))dF(y) \\
&+ \int_x^{\overline{\psi}(x)} (v - \beta(y) - (1-\alpha)(x - \beta(y)))dF(y),
\end{aligned} \tag{37}
$$

$$
\begin{aligned}
0 = &\phi'(v)(v - (1-\alpha)\beta(v) - \alpha\beta(\phi(v)))f(\phi(v)) \\
&- (1-\alpha)\beta'(v)F(\phi(v)) \\
&+ \alpha(v - \beta(v))\left[\frac{\overline{\psi}'(v)f(\overline{\psi}(v))}{2} - f(v)\right] \\
&- (1-\alpha)(F(\overline{\psi}(v)) - F(v)).
\end{aligned} \tag{38}
$$

Equations (36) and (38) are delayed differential equations that cannot be solved analytically in general.[15] Therefore, we apply numerical methods to compute approximate solutions for particular parameter values. The numerical algorithm is summarized in Appendix B and spelled out in detail in a supplementary document which is available for download (Lengwiler and Wolfstetter, 2005a).

While the theory of delayed differential equations guarantees existence of some solution (Kuang, 1993, Theorem 2.1), it does not guarantee existence of a monotone solution.[16] Numerical methods can never prove existence because they give us an approximate solution at best. In our numerical analysis we consider various combinations of n and α, but of course we must exclude (n, α) for which no equilibrium exists. We therefore begin with a necessary condition for existence, which restricts the set of parameters.

In our numerical analysis, we assume uniformly distributed valuations. Suppose β is a solution of (36) for the uniform distribution, and let $s := \beta'(0)$. Consider the first-order Taylor approximation of β at 0. Then $\beta(v) \approx sv$, $\phi(v) \approx 2s/(1+s)$,

[15] An exception is $\alpha = 1$, in which case truthful bidding is an equilibrium. To see why, note that $\phi(v) = v$ and $\overline{\psi}(v) = v$ if $\beta(v) = v$, by (28) and (33), respectively. Use this fact, set $\alpha = 1$, and check that $\beta(v) = v$ solves (36) and (38). Similarly, for arbitrary fixed α, β converges pointwise to truthful bidding as n grows indefinitely. Formally, $\forall v \; \lim_{n\to\infty} \beta(v) = v$. The argument is again very similar. Observe that $\phi(v) \to v$ and $\overline{\psi}(v) \to v$ as $\beta(v) \to v$. Use this fact and also observe that $\lim_{n\to\infty} G(v) = 0$ for all $v < 1$. With this information, one checks that $\beta(v) = v$ solves (36) as $n \to \infty$.

[16] β is not necessarily differentiable at some points. For instance, it may have a kink at the smallest v where $\overline{\psi}(v) = 1$, so both, left-hand and right-hand derivatives, exist, but do not coincide. However, this does not cause any problem, because the derivative in the delay differential equation is defined as the right-hand derivative.

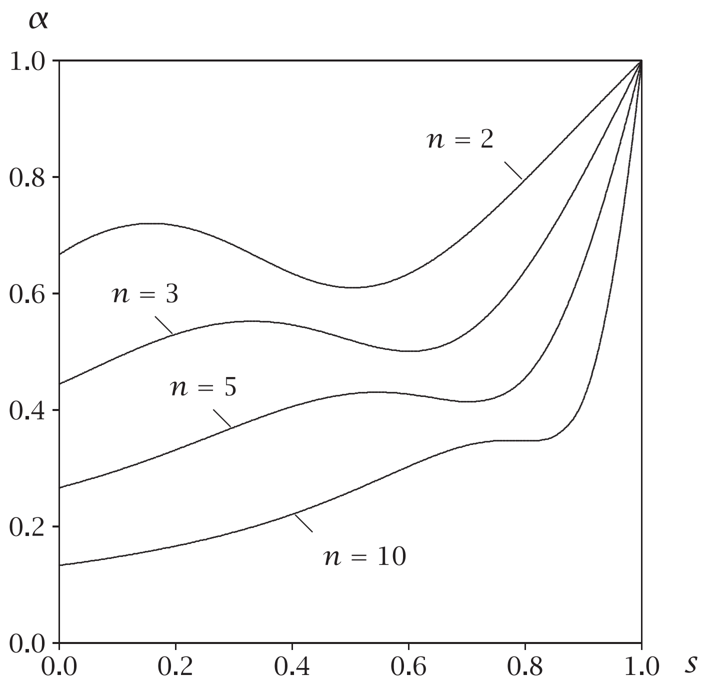

Figure 2: Roots of the polynomial (39) for various n.

and $\overline{\psi}(v) \approx (1+s)/(2s)$ for $v \approx 0$. Therefore, s solves the polynomial

$$\begin{aligned} f(s,\alpha,n) :=\ & \frac{2s}{1+s}\left(1-(1-\alpha)s-\alpha s\frac{2s}{1+s}\right)(n-1)\left(\frac{2s}{1+s}\right)^{n-2} \\ & -(1-\alpha)s\left(\frac{2s}{1+s}\right)^{n-1}+\alpha(1-s)(n-1)\left(\frac{1+s}{4s}-1\right) \\ & -(1-\alpha)(n-1)\left(\frac{1+s}{2s}-1\right) \\ & +\alpha(n-1)(n-2)\left(\frac{1+s}{2s}-1-\frac{s}{2}\left(\left(\frac{1+s}{2s}\right)^2-1\right)\right). \end{aligned} \tag{39}$$

Figure 2 plots combinations of s and α for which $f(s,\alpha,n)=0$. Notice that, for given n, there is no real root in the unit interval if α is too small. Define $\alpha^*(n) := \inf\{\,\alpha : \exists s \in [0,1]\ f(s,\alpha,n)=0\,\}$. An equilibrium cannot exist if $\alpha < \alpha^*(n)$.

Figure 3 depicts numerical solutions of the equilibrium bid functions. For $n=2$, the necessary condition $\alpha \geqslant \alpha^*(2) = 0.6101$ seems to be also sufficient. This is not true for larger n; for instance, $\alpha^*(5) = 0.2667$, yet we find numerical solutions of (36) with small error only for $\alpha \geqslant 0.4515$.

Figure 4 depicts the equilibrium allocations in the state space for $\alpha = 0.8$ and for $\alpha = 0.6101$ (the smallest α for which we have found a solution). Bidder 1 wins the object in the shaded area; in the white area, bidder 2 is the winner. Efficiency requires that bidder 1 wins if and only if he has the higher valuation, $v_1 > v_2$, and *vice versa*. Therefore, for efficiency, the entire area below the 45°-line should be shaded, and the entire area above it should be white. Clearly, the equilibrium allocation is not efficient. In both parameter cases, there is a white wedge in the

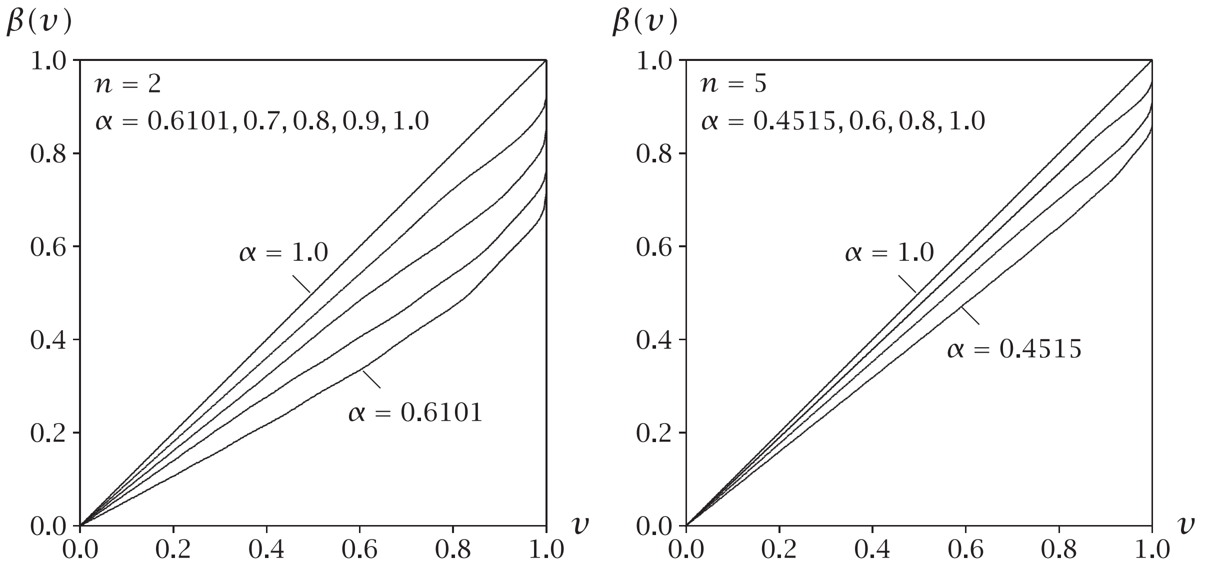

Figure 3: Approximate equilibrium bid functions for the uniform distribution, for $n = 2$ (left panel) and $n = 5$ (right panel) and various values for α.

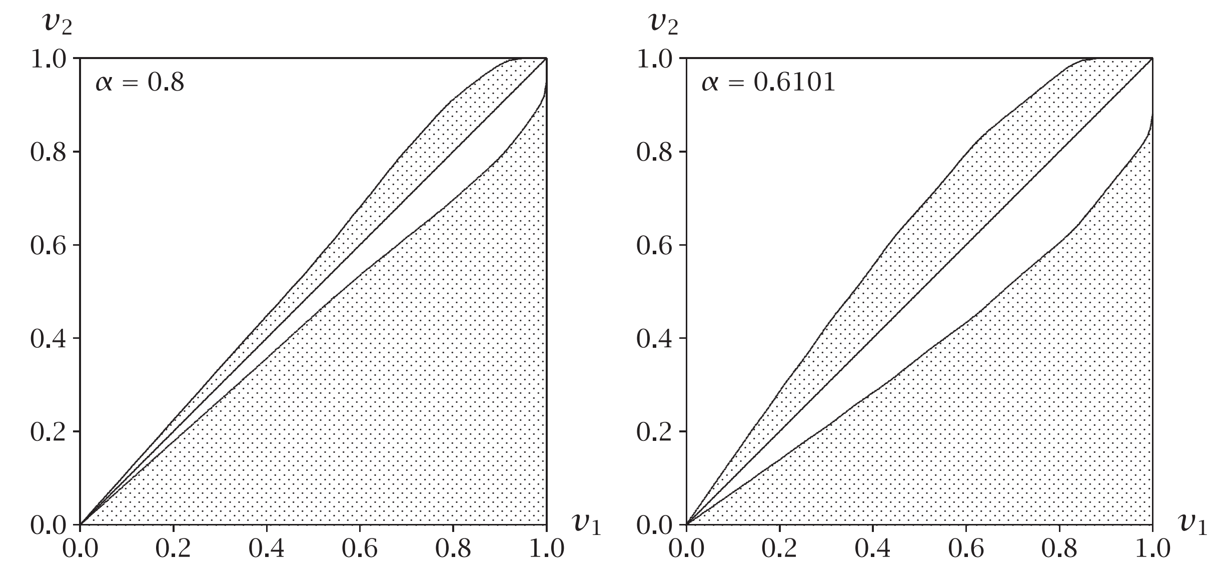

Figure 4: Equilibrium allocation for $n = 2$ and two different values of α. Bidder 1 wins in the shaded regions, bidder 2 in the white regions.

area below the 45°-line that should be shaded (there, bidder 2 wins although he has the lower valuation), and a shaded wedge in the area above the 45°-line that should be white (there, bidder 1 wins although he has the lower valuation). These "wedges" indicate the presence of type II corruption which changes the allocation by letting the second highest bidder win the auction.

Comparing the two figures for $\alpha = 0.6101$ and $\alpha = 0.8$ illustrates the second fact that the two inefficiency "wedges" increase in size as α is lowered. The intuition for this is as follows: type II corruption requires bid shading. If α is close to one, the equilibrium bid function is close to truthful bidding, therefore there is

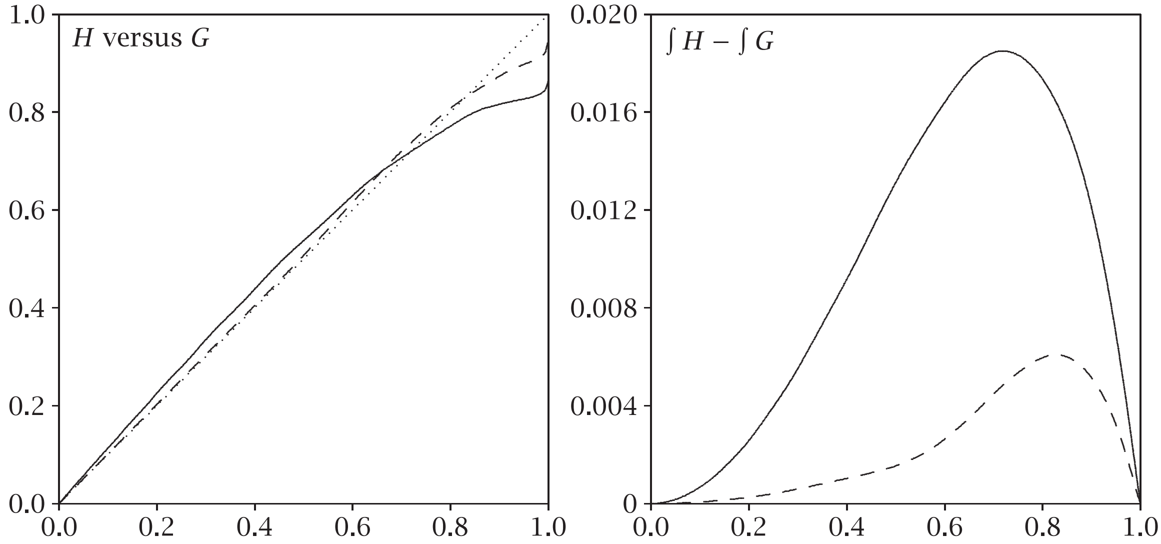

Figure 5: *Left panel*: winning probabilities for two bidders ($n = 2$) in the efficient auction (G, dotted line) and in the first-price auction with both types of corruption (H), with $\alpha = 0.6101$ (solid line) and $\alpha = 0.8$ (dashed line). *Right panel*: effect of corruption on the expected payoffs of bidders, as function of their valuation, compared to the efficient auction.

almost no room for type II corruption. As α is lowered, bids are shaded more, and this, in turn, makes more room for type II corruption, by increasing the spread of valuations, $v_1 - v_2$, for which the auctioneer benefits the most from allowing the second highest bidder to win.

Finally, we assess the welfare and distribution impact of corruption. A bidder with valuation v wins the first-price auction with probability $H(v) := G(\overline{\psi}(v)) - G(v) + G(\phi(v))$. This differs from the efficient allocation rule $G(v)$. This difference is the source of inefficiency.

In all our computations, H and G differ almost everywhere, yet H is still a monotone increasing function. Therefore, by Myerson (1981, Lemma 2), bidders' expected payoff can be computed as

$$u(v) := \int_0^v H(y)dy = \int_0^v (G(\overline{\psi}(v)) - G(v) + G(\phi(v)))dy. \tag{40}$$

Figure 5 plots H, G, and bidders' gain from corruption, $\int_0^v (H(y) - G(y))dy$, for $n = 2$, two values of α, and uniformly distributed valuations, on the basis of the numerical computations of β. All bidders, except the two extreme types $v = 0$ and $v = 1$, benefit from corruption. Surprisingly, they benefit *more* if the winning bidder's share of the gain from corruption, α, is reduced. These properties carry over to larger n.

The auctioneer receives a share $(1 - \alpha)$ of the gain from corruption, which is either the difference of the two highest bids in the event of type I corruption, or the difference between the second highest valuation and the highest bid in the

Table 1: Changes of welfare and payoffs due to corruption.

$n = 2$	α	welfare	seller	auctioneer
	0.9	−0.001 (−0.2%)	−0.032 (−9.6%)	+0.030
	0.8	−0.005 (−0.7%)	−0.063 (−18.9%)	+0.054
	0.7	−0.011 (−1.6%)	−0.097 (−29.0%)	+0.074
	0.6101	−0.019 (−2.9%)	−0.134 (−40.1%)	+0.095
$n = 3$	α	welfare	seller	auctioneer
	0.9	−0.001 (−0.1%)	−0.025 (−5.0%)	+0.024
	0.8	−0.002 (−0.3%)	−0.050 (−10.0%)	+0.045
	0.7	−0.006 (−0.8%)	−0.077 (−15.4%)	+0.065
	0.6	−0.011 (−1.5%)	−0.109 (−21.8%)	+0.086
	0.5010	−0.019 (−2.5%)	−0.150 (−30.0%)	+0.112
$n = 5$	α	welfare	seller	auctioneer
	0.9	−0.000 (−0.0%)	−0.017 (−2.6%)	+0.016
	0.8	−0.001 (−0.2%)	−0.035 (−5.2%)	+0.032
	0.7	−0.003 (−0.4%)	−0.054 (−8.1%)	+0.047
	0.6	−0.006 (−0.8%)	−0.076 (−11.4%)	+0.063
	0.5	−0.011 (−1.3%)	−0.104 (−15.6%)	+0.082
	0.4515	−0.013 (−1.6%)	−0.120 (−18.0%)	+0.094
$n = 10$	α	welfare	seller	auctioneer
	0.9	−0.000 (−0.0%)	−0.012 (−1.4%)	+0.009
	0.8	−0.001 (−0.1%)	−0.021 (−2.6%)	+0.018
	0.7	−0.002 (−0.2%)	−0.032 (−3.9%)	+0.026
	0.6	−0.003 (−0.4%)	−0.045 (−5.4%)	+0.036
	0.5	−0.005 (−0.6%)	−0.060 (−7.3%)	+0.047
	0.4013	−0.008 (−0.9%)	−0.080 (−9.7%)	+0.062

event of type II corruption. His expected payoff is therefore

$$\pi_{\text{auc}} := (1-\alpha)\int_0^1 \left(\int_0^{\phi(x)} (\beta(x) - \beta(y)) f_{X_2X_1}(y,x)\,dy + \int_{\phi(x)}^{x} (y - \beta(x)) f_{X_2X_1}(y,x)\,dy \right) dx, \tag{41}$$

where $f_{X_2X_1}(y,x) := n(n-1)F(y)^{n-2}f(y)f(x)$, $y \leqslant x$, denotes the joint density of the highest and second highest of a sample of n valuations.

Similarly, the payoff of the seller depends on the distance between the two highest bids, because this distance determines which type of corruption is being played,

$$\pi_{\text{seller}} := \int_0^1 \left(\int_0^{\phi(x)} \beta(y) f_{X_2X_1}(y,x)\,dy + \int_{\phi(x)}^{x} \beta(x) f_{X_2X_1}(y,x)\,dy \right) dx. \tag{42}$$

Finally, recall that inefficiency occurs only in the event of type II corruption. Therefore, the welfare loss of corruption is

$$\pi_{\text{loss}} := \int_0^1 \int_{\phi(x)}^{x} (x-y) f_{X_2X_1}(y,x)\,dy. \tag{43}$$

Table 1 summarizes how welfare and payoffs change compared to the equilibrium in the absence of corruption, based on the above formulas and the numerical computations assuming a uniform distribution. The results contained in this table, together with our results on bidders' payoffs, indicate that both — the auctioneer and all bidders — benefit from low competition and small α. This suggests that corruption is hard to fight as both involved parties benefit from it.

APPENDIX

A SECOND-ORDER CONDITIONS

When we derived the equilibria of various auction games, we employed only the first order conditions for a best reply. Here we complete the proofs of equilibrium. We show that our candidate equilibrium strategies are indeed mutual best replies.

LEMMA 1 *Consider a strict monotone increasing strategy β. That strategy is a symmetric (strict) Bayesian Nash equilibrium if*

$$\frac{\partial}{\partial x} U(v,x)|_{x=v} = 0 \quad \forall v, \tag{44}$$

$$\frac{\partial^2}{\partial x \partial v} U(v,x) > 0 \quad \forall v, x. \tag{45}$$

PROOF: Consider a bidder with valuation v who assumes that all rival bidders play the strict monotone increasing candidate equilibrium strategy β. Without loss of generality, that bidder considers only deviating bids, $\beta(x)$, for $x \in [0,1]$.

Now consider deviating bids, $\beta(x)$. Then, by (44) and (45),

$$x < v \Rightarrow \frac{\partial}{\partial x} U(v,x) > \frac{\partial}{\partial x} U(v,x)|_{v=x} = 0 \tag{46}$$

$$x > v \Rightarrow \frac{\partial}{\partial x} U(v,x) < \frac{\partial}{\partial x} U(v,x)|_{v=x} = 0. \tag{47}$$

This proves that $U(v,x)$ is increasing in x for all $x < v$, and decreasing for all $x > v$. Therefore, bidding according to the candidate equilibrium strategy, $\beta(v)$ is a (strict) best reply; hence, β is a (strict) equilibrium. □

The equilibrium strategies proposed in the present paper are strict monotone increasing and, by construction, satisfy (44). Since

$$\frac{\partial^2}{\partial x \partial v} U(v,x) = G'(x) > 0, \tag{48}$$

(45) is also satisfied. We conclude that Lemma 1 applies; therefore, our candidate equilibria are (strict) Bayesian Nash equilibria.

B NUMERICAL COMPUTATIONS

Throughout the numerical exercise we assume uniformly distributed valuations. We search for solutions of the differential equation for the first-price auction (36). To make this problem suitable for numerical analysis we search for approximate solutions within the class of strictly increasing, continuous, piecewise linear functions that satisfy the boundary condition $\beta(0) = 0$. Let $G := \{0, 1/g, \ldots, (g-1)/g, 1\}$, for some $g \in \mathbb{N}$, be a uniform grid on the unit interval. An increasing, piecewise linear function is defined by the numbers $\{\beta(v) : v \in G\}$. This transforms the infinite-dimensional problem (36) into a g-dimensional problem. For the computations we set $g = 200$.

We explore several root finding methods and step-size optimization techniques.[17] Our computations reveal that the steepest descent method works up to a point, but becomes prohibitively slow as we approach the solution. The Gauss-Newton method, on the other hand, works well if we start close from the solution, but is very demanding with regard to the choice of the initial guess. The Levenberg-Marquard method is an algorithm that combines the two strengths of steepest descent and Gauss-Newton, and our computations reveal that this method is very efficient when applied to our problem. It finds the solution within a reasonable number of iterations and produces small errors.

The $C^\sharp$ source code, a detailed description of the program and of the algorithms we use, together with some supplementary results, are available for download, see Lengwiler and Wolfstetter (2005a).

References

Arozamena, L., and F. Weinschelbaum (2004): "The Effect of Corruption on Bidding Behavior in First-Price Auctions," Discussion paper, Universidad de San Andrés.

Burguet, R., and Y.-K. Che (2004): "Competitive Procurement with Corruption," *Rand Journal of Economics*, 35, 50–68.

Burguet, R., and M. Perry (2003): "Bribery and Favoritism by Auctioneers in Sealed-Bid Auctions," Discussion paper, Institute of Economic Analysis, UAB, Barcelona.

Celantani, M., and J.-J. Ganuza (2002): "Corruption and Competition in Procurement," *European Economic Review*, 46, 1273–1303.

Compte, O., A. Lambert-Mogiliansky, and T. Verdier (2005): "Corruption and Competition in Public Market Auctions," *Rand Journal of Economics*, 35, (forthcoming).

Cramton, P. C., R. Gibbons, and P. Klemperer (1987): "Dissolving a Partnership Efficiently," *Econometrica*, 55, 615–632.

[17] A good introduction to numerical methods is Heath (2002).

DAVID, H. A. (1970): *Order Statistics*, Wiley Series in Probability and Mathematical Statistics. Wiley.

ENGELBRECHT-WIGGANS, R. (1994): "Auctions with Price-Proportional Benefits to Bidders," *Games and Economic Behavior*, 6, 339–346.

GÜTH, W. (1995): "Preisregeln für Auktionen und Ausschreibungen: Eine ordnungspolitische Analyse," *Zeitschrift f. Wirtschafts- und Sozialwissenschaften*, 115, 1–26.

HEATH, M. T. (2002): *Scientific Computing. An Introductory Survey*. McGraw-Hill, 2nd edn.

JENSEN, I. (1984): *Quellen und Zeugnisse zur Druckgeschichte von Goethes Werken*. Akademie-Verlag.

KOC, S., AND W. NEILSON (2005): "Bribing the Auctioneer in First-Price Sealed-Bid Auctions," Discussion paper, Texas A&M University.

KUANG, Y. (1993): *Delay Differential Equations with Applications in Population Dynamics*, Mathematics in Science and Engineering. Academic Press.

LAFFONT, J.-J., AND J. TIROLE (1991): "Auction Design and Favoritism," *International Journal of Industrial Organization*, 9, 9–42.

LENGWILER, Y., AND E. WOLFSTETTER (2000): "Auctions and Corruption," Working Paper 401, CESifo (revised 2002 and 2004).

——— (2005a): "Notes on the Numerical Computations in 'Bid Rigging — An Analysis of Corruption in Auctions'," Discussion paper, University of Basel and Humboldt University at Berlin, www.wwz.unibas.ch/witheo/yvan/research.html.

——— (2005b): "On some Auction Rules for Amicable Divorce in Equal Share Partnerships," Discussion paper, Humboldt University at Berlin.

MCAFEE, R., AND J. MCMILLAN (1992): "Bidding Rings," *American Economic Review*, 82, 579–599.

MENEZES, F., AND P. MONTEIRO (2003): "Corruption and Auctions," Discussion paper, Australian National University.

MOLDOVANU, B., AND M. TIETZEL (1998): "Goethe's Second-Price Auction," *Journal of Political Economy*, 106, 854–859.

MYERSON, R. B. (1981): "Optimal Auction Design," *Mathematics of Operations Research*, 6, 58–73.

RILEY, J. G. (1989): "Expected Revenue from Open and Sealed Bid Auctions," *Journal of Economic Perspectives*, 3, 41–50.

NOTES ON THE NUMERICAL COMPUTATIONS IN "BID RIGGING — AN ANALYSIS OF CORRUPTION IN AUCTIONS"

YVAN LENGWILER
University of Basel
Dept. of Economics (WWZ)
Petersgraben 51
CH–4003 Basel
Switzerland
yvan.lengwiler@unibas.ch

ELMAR WOLFSTETTER
Humboldt University at Berlin
Institute of Economic Theory I
Spandauer Str. 1
D–10178 Berlin
Germany
wolfstetter@wiwi.hu-berlin.de

April 2005

Contents

1 THE PROBLEM

We want to find the solution to the delayed differential equation of the first-price auction. For the numerical exercise, we assume a uniform distribution of the valuations throughout. The differential equation then simplifies to

$$\begin{aligned} D(v) := {} & \phi'(v)(v - (1-\alpha)\beta(v) - \alpha\beta(\phi(v)))(n-1)\phi(v)^{n-2} \\ & - (1-\alpha)\beta'(v)\phi(v)^{n-1} \\ & + \alpha(v - \beta(v))(n-1)v^{n-2}\left[\frac{\overline{\Psi}'(v)}{2} - 1\right] \\ & - (1-\alpha)(n-1)v^{n-2}(\overline{\Psi}(v) - v) \\ & + \alpha(n-1)(n-2)v^{n-3}\left[\left(\overline{\Psi}(v) - v\right)v - \int_v^{\overline{\Psi}(v)} \beta(y)dy\right] = 0. \end{aligned} \tag{1}$$

Note that the evaluation of (1) is not completely trivial because we first need to determine $\overline{\Psi}(v)$ and $\phi(v)$. $\overline{\Psi}$ involves the inverse of β, so we will have to make sure that our candidate β is always invertible. ϕ demands a little more. It is the solution of a fixed point problem,

$$\phi(v) + \beta(\phi(v)) - 2\beta(v) = 0. \tag{2}$$

Because the left-hand side of this equation is monotonic in $\phi(v)$, and is negative if $\phi(v) = 0$, and positive if $\phi(v) = v$ (as long as there is bid shading, $\beta(v) < v$), we can use the bi-section method and determine $\phi(v)$ to an arbitrary precision. The required precision is determined in the code by `phiEps` and set by default to machine precision. For the remaining, we draw heavily from the splendid book by Heath (2002).

2 ROOT FINDING METHODS

To make problem (1) suitable for numerical analysis we search for approximate solutions within the class of strictly increasing, continuous, piecewise linear functions that satisfy the boundary condition $\beta(0) = 0$. Let $G := \{0, 1/g, \dots, (g-1)/g, 1\}$, for some $g \in \mathbb{N}$, be a uniform grid on the unit interval. A piecewise linear function is defined by the numbers $\{\beta(v) : v \in G\}$. This transforms the infinite-dimensional problem (1) into a g-dimensional problem. You can select the fineness of the grid on valuation space by changing the constant `g` in the source code.

All iterative procedures start from some initial "guess" (more on this later). Let β_0 be the initial bid function, and $\beta_1, \beta_2, \dots$ denote the bid functions along an iteration. A root finding algorithm is a rule that prescribes how to get from β_i to β_{i+1}. The idea is to do this in such a way that the sequence of β_i approaches the true solution in the process. We have implemented several standard methods for such problems. You can select the method by setting the switch `rootmethod` to a value between `0` and `3`.

2.1 *The Gauss-Newton method*

A standard method for moving from β_i to β_{i+1} is the Gauss-Newton method. It involves the Jacobian of D_i with respect β_i, which we denote with J_i. To approximately compute the components of the Jacobian, for each $v_k := k/g \in G$, we compute D_i at two points, namely β_i^+ and β_i^-, given by $\beta_i^+(v_k) := \beta_i(v_k) + \delta$, $\beta_i^-(v_k) := \beta_i(v_k) - \delta$, and $\beta_i^+(v) = \beta_i^-(v) = \beta_i(v)$ for all $v \neq v_k$.[1] The Jacobian is then approximately given by

$$J \approx \begin{bmatrix} \frac{D_1(\beta_1^+)-D_1(\beta_1^-)}{2\delta} & \cdots & \frac{D_1(\beta_g^+)-D_1(\beta_g^-)}{2\delta} \\ \vdots & \ddots & \vdots \\ \frac{D_g(\beta_1^+)-D_g(\beta_1^-)}{2\delta} & \cdots & \frac{D_g(\beta_g^+)-D_g(\beta_g^-)}{2\delta} \end{bmatrix}.$$

The iteration proceeds according to the following rule

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i J_i^{-1} D_i. \tag{3}$$

τ_i is a positive number called the step size. As explained later, we optimize over the step size τ_i.

2.2 *Steepest descent method*

Another standard method is the method of steepest descent. Compute the sum of squared errors,

$$\mathrm{SSE}_i := \frac{1}{2} \sum_{v \in G} D_i(v)^2,$$

(the division by 2 is a normalization that will make sense later). We aim at minimizing SSE. To that avail we compute the gradient of SSE with respect to $\{\beta_i(v) : v \in G\}$. As before, we approximate the gradient by computing a discrete difference of the components of the bid function ($\pm\delta$): we compute $\mathrm{SSE}_i^+(\beta_i, v)$ as the SSE that would result if $\beta_i(v) \mapsto \beta_i(v) + \delta$ for a given $v \in G$, and $\mathrm{SSE}_i^-(\beta_i, v)$ as the SSE that would result if $\beta_i(v) \mapsto \beta_i(v) - \delta$. The gradient is then approximately equal to

$$\nabla \mathrm{SSE}(\beta_i) := \left(\frac{\mathrm{SSE}_i^+(\beta_i, v) - \mathrm{SSE}_i^-(\beta_i, v)}{2\delta} \right)_{v \in G}.$$

With this information, we can iterate according to the following rule

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i \nabla \mathrm{SSE}(\beta_i). \tag{4}$$

We move here into the direction in which SSE decreases the fastest locally, hence the name of the method. τ_i is again the step size, which we optimize.

[1] δ is called `diffDelta` in the program, and is equal to `diffDelta = 1E-10` by default; you are free to change this parameter.

2.3 *A hybrid method*

The Gauss-Newton method finds the minimum of a quadratic function in one iteration. For non-quadratic problems, the initial point has to be sufficiently close to the solution. If not, the method might not converge at all. So this method is fast if we are close to the solution, but quite demanding in terms of choice of starting point.

Compared to the Gauss-Newton method, the steepest descent method is slow. Like the Gauss-Newton method, it exhibits global convergence for quadratic problems (though not in one iteration step). The advantage of steepest descent over the Gauss-Newton method is that steepest descent is much less demanding with respect to the starting point. Even if we are far from the solution, the method will point into more or less the right direction[2] and initially converge at a decent speed. Only when we get close to the solution does convergence speed deteriorate.

These observations suggest a hybrid method, that combines the advantages of the steepest descent method in the early phase of the process, and later switches to the Gauss-Newton method. We switch from steepest descent to Gauss-Newton when there has been no significant improvement (`switchEps = 1E-12`) over sufficiently many consecutive iterations (`switchKeep = 5`).

2.4 *The Levenberg-Marquardt method*

This method is similar in spirit to the hybrid method we have just discussed, but instead of completely switching from one method to the other, it moves more gradually between the two. The iteration rule is given by

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i \left((1-\lambda_i)J_i^T J_i + \lambda_i I\right)^{-1} J_i^T D_i, \qquad \lambda_i \in (0,1). \tag{5}$$

Note that (5) is a convex combination between the Gauss-Newton rule (3), when $\lambda_i = 0$, and the steepest descent rule (4), when $\lambda_i = 1$.[3]

The strategy of the method is to dynamically change λ_i in a smart fashion. Define $\mu_i := \lambda_i(1-\lambda_i)^{-1}$. We start from some initial μ_0 (`muInit = 1E-3`). If the SSE improves from one iteration to the next, we decrease μ by dividing it by some factor $m > 0$ (in the code, m is called `muFactor = 10`), thus moving closer to the Gauss-Newton method; if the SSE deteriorates we increase μ_i by multiplying it with m, thus moving into the direction of the steepest descent method.

We slightly vary this strategy because we have found that it works better. We decide about increasing or decreasing μ_i not by comparing the SSE from one iteration to the next. Instead, we decrease μ_i only if the current SSE is better than the best SSE that has been encountered so far during the whole process; we increase it otherwise.

[2] Of course, we might get stuck on a local minimum if SSE is not unimodal.

[3] Note that $\nabla \text{SSE} = J^T D$. Here is where the division by 2 in the definition of SSE comes in.

3 OPTIMIZING THE STEP SIZE

Optimization of the step size is quite important, particularly as long as we are far from the solution. Moreover, our root finding methods might suggest a step that would make the next bid function β_{i+1} locally decreasing or that might violate weak bid shading. Non-monotonicity in particular would cause trouble in further calculations by making the bid function not invertible. We proceed as follows: in each iteration step, we first determine the maxmimum size of τ that still guarantees weak monotonicity and bid shading; call this step size τ^*. Then we optimize by searching for a step τ in the interval $(0, \tau^*)$ which minimizes SSE.[4]

A good algorithm for finding the minimum along a line if the function is unimodal is golden section search. You can select this option by setting `taumethod = 1` along with the required precision (`tauEps`).

Because we cannot be sure that SSE is unimodal in our search direction, we have also implemented a more expensive procedure (`taumethod = 0`), which we call "brute force": we compute SSE for step sizes $\tau \in \{\tau^* \, s \, S^{-1} : s = 1, \ldots S-1\}$, and we select the best one. S is called `tausteps` in the program. Note that we restrict $0 < \tau^* S^{-1} \leqslant \tau \leqslant \tau^*(S-1)S^{-1} < \tau^*$. $\tau = 0$ would immediately lead to an infinite loop; the algorithm would be stuck. $\tau = \tau^*$ is very likely to lead to a locally flat bid function. This makes inverting the bid function impossible, and also raises the possibility that τ^* would become zero in the next iteration step. The algorithm would again be stuck. Yet, the strategy of disallowing extreme step sizes is only partially successful. It happens on rare occasions that τ^* becomes smaller than machine precision. The algorithm stops in such a case.

If we are close to the solution, it can be that the smallest step is still too large so that the algorithm would overshoot. If the SSE is larger even with the smallest allowed step, we re-optimize the step size by evaluating the SSE at steps $\tau \in \{\tau^* \, s \, S^{-k} : s = 1, \ldots S-1\}$, with $k = 2$. We keep increasing k until the iteration leads to an improvement of the SSE. However, in no case do we allow τ to become smaller than $\tau^* \mathrm{SSE}/(g+1)$. This limit allows smaller steps once the error is small, but avoids getting stuck on very small steps as long as the error is still large.

4 FINDING AN INITIAL GUESS

4.1 *Starting from a linear bid function*

Because we cannot hope to have global convergence, the choice of the initial bid function β_0 has to be done with care, and we should try to start from a point which is as close as possible to the solution. A simple idea is to start from the solution of a simpler problem

[4] If this requirement does not put a constraint on the step size, we allow τ^* to be at most equal to `GRANDMAXTAU = 2.0`, so the bid function is changed by at most twice as much as the step size suggested by the chosen method.

which we can solve explicitly. We start from the solution of the restricted first-price auction with type I corruption only. The rationale for this choice is the hope that adding the possibility of type II corruption does not alter the solution in a too extreme fashion. With the uniform distribution of valuations, this bid function is linear and given by

$$\beta_0(v) := \frac{n-1}{n-\alpha}\,v. \tag{6}$$

More generally, you can choose to start from any linear bid function by setting `initmethod = 0`. The slope of this initial bid function is set in the variable `slope`, which is by default set to the slope defined in (6).

4.2 *Starting from a non-linear bid function*

One could also start from a non-linear bid function. To do that, you need to write an alternative to the procedure `InitLinear`, maybe along these lines:

```
private static void InitPower()
{
    Console.WriteLine(">>> Filling in power bid function " +
                      "as an initial guess...");
    for (int j=1; j<=g; j++)
        betavec[j, 0] = slope * Math.Pow(val(j),exponent);
    compute();
}
```

You will have to declare the constant `exponent` in the beginning of the code somewhere (preferably in the neighborhood of the declaration of `slope`), and also change a part of the `Main()` method as follows,

```
switch (initmethod)
{
    case 0:
       InitLinear();
       break;
    case 1:
       RunGridSearch();
       OutputResult();
       break;
    case 2:
       InitPower();
       break;
}
```

Setting `initmethod = 2` would then select the power bid function as the starting point.

4.3 *Grid search*

An alternative to a fixed initial guess is to run a grid search (select `initmethod = 1`). In a grid search, we determine the RMSE of a large number of bid functions and select the one with the smallest RMSE as the starting point. More precisely, for the first dimension $i = 1$ (that is, $v_1 = 1/g$), we select Q equally spaced bids b_1 strictly between 0 and v_1 (Q is called `GridSearchSteps` in the code). For each of these bids, we then select Q bids for the second dimension, again equally spaced, and strictly between b_1 and v_2, in order to observe strictly monotonicity and bid shading. We do this through all g dimensions. This gives rise to Q^g bid functions. It is obvious that with a reasonably fine grid on the valuation space, say $g = 50$, this kind of grid search is not feasible because already with $Q = 2$ we would have to evaluate more than 10^{15} bid functions. So grid search is feasible only if we work with a coarse grid on the valuations, at least initially.

4.4 *A variation: progressively finer grid method*

Since it is important to start from an initial bid functions which is close to the final solution, and it is easier to find the root for a problem with less dimensions rather than more, one can follow a strategy of starting with a relatively coarse grid on the valuation space, and making the grid progressively finer. To consider an extreme example, suppose we start with $g = 1$. This amounts to searching for a solution within the class of linear functions. Apply a root finding method as described before. Then subdivide the grid on the valuation space, say by 4. So now $g = 4$ and we linearly interpolate the bid function for the new points in the valuation grid. We then again apply one of the root finding methods, and again subdivide the valuation space. We keep on doing this until we reach a reasonable fineness of the grid on the valuation space. This method has potentially two advantages. Firstly, it allows us to start with a very coarse grid, making the grid search a feasible choice for the initial guess. Secondly, after each subdivision of the valuation space, we start from a bid function which should be rather close to the solution for the new valuation grid.

In the code, the progressively finer grid method is controlled by two parameters: `NbSubdivisions` is the number of rounds in which the valuation grid is made progressively finer, and `SubdivisionFactor` is the factor by which it is made finer in each round. Thus, if we start with a grid of $g = 3$, which we subdivide twice with a factor of four, we end up with $3 \times 4^2 = 48$ points in the valuation space. Setting `NbSubdivisions = 0` turns off the progressively finer grid method.

5 THE RESULTS

We have not made very good experiences with the progressively finer grid method, so we do not use it (`NbSubdivisions = 0`). Instead we run a rather fine grid on the valuation space to begin with (`g = 200`). This rules out the grid search for the starting point (`initmethod = 0`), and we use a linear function following (6).

We find that the Levenberg-Marquardt method is the best choice for our problem. An iteration step is quite slow, but it makes up for this drawback with fast convergence. If there is an equilibrium, it finds it within a small number of iterations. The steepest descent method works also, although convergence is very slow and it does not seem to generate the same amount of precision.

For the optimization of the step size, we find that golden section search works well, so we use it throughout. The brute force method often works too, but is somewhat inferior.[5]

Finally, we stop the iteration if no improvement has occurred for sufficiently many (`keepiter = 20`) iterations, or after a maximum of `maxiter = 100` iterations. We then report the iteration step with the smallest SSE that we have detected so far.

Figure 1 depicts the approximate equilibrium bid functions; Table 1 reports the precision we achieve with these computations. For each n we have investigated, there is a borderline α at which the SSE increases dramatically when we increase α by just 0.0001 beyond this limit. These boderline αs are identified in the table.

6 THE PROGRAM

An explanation about our choice of platform is in order. The program is essentially a simple procedure, a list of commands that is executed in a given sequence. Given this fact, it may seem surprising that we chose $C^\sharp$ for implementation. Object orientation — one of the more important aspects of $C^\sharp$ — is not important for our task, and is in fact not used. So it appears that $C^\sharp$ is an odd choice for such a project. It would be more natural to implement it with Gauss or Matlab. The decisive advantage of the .NET Framework (and the $C^\sharp$ compiler that comes with it) is that it is free. As a consequence, everyone (with a Windows computer and internet access) can run our program without spending a nickel or wasting time trying to get spending approved by the University bureaucracy. To use the $C^\sharp$ code, you need Microsoft's .NET Framework SDK Version 1.1, which is available from `http://msdn.microsoft.com/netframework/`. You may find it comfortable to use an IDE for changing the program (we did, at least). The standard IDE for the .NET Framework is Microsoft's Visual Studio .NET, which is very good but also very expensive. A free alternative can be downloaded from `http://www.icsharpcode.net/OpenSource/SD/`.

A ZIP archive containing the program and computed results is available for download from our websites (at the time of this writing, `http://www.wwz.unibas.ch/witheo/yvan/research.html`). The archive contains this description as a PDF file. In addition, it contains a file called `Main.cs`, which is the $C^\sharp$ source code, and an Excel file called

[5]The parameters are `tauEps = 1E-2` for the golden section search and `tausteps = 10` for the brute force method. Brute force step size optimization often seems to work more harmoniously with the steepest descent root finding method, but since we do not use steepest descent, we have also no need for brute force step size optimization.

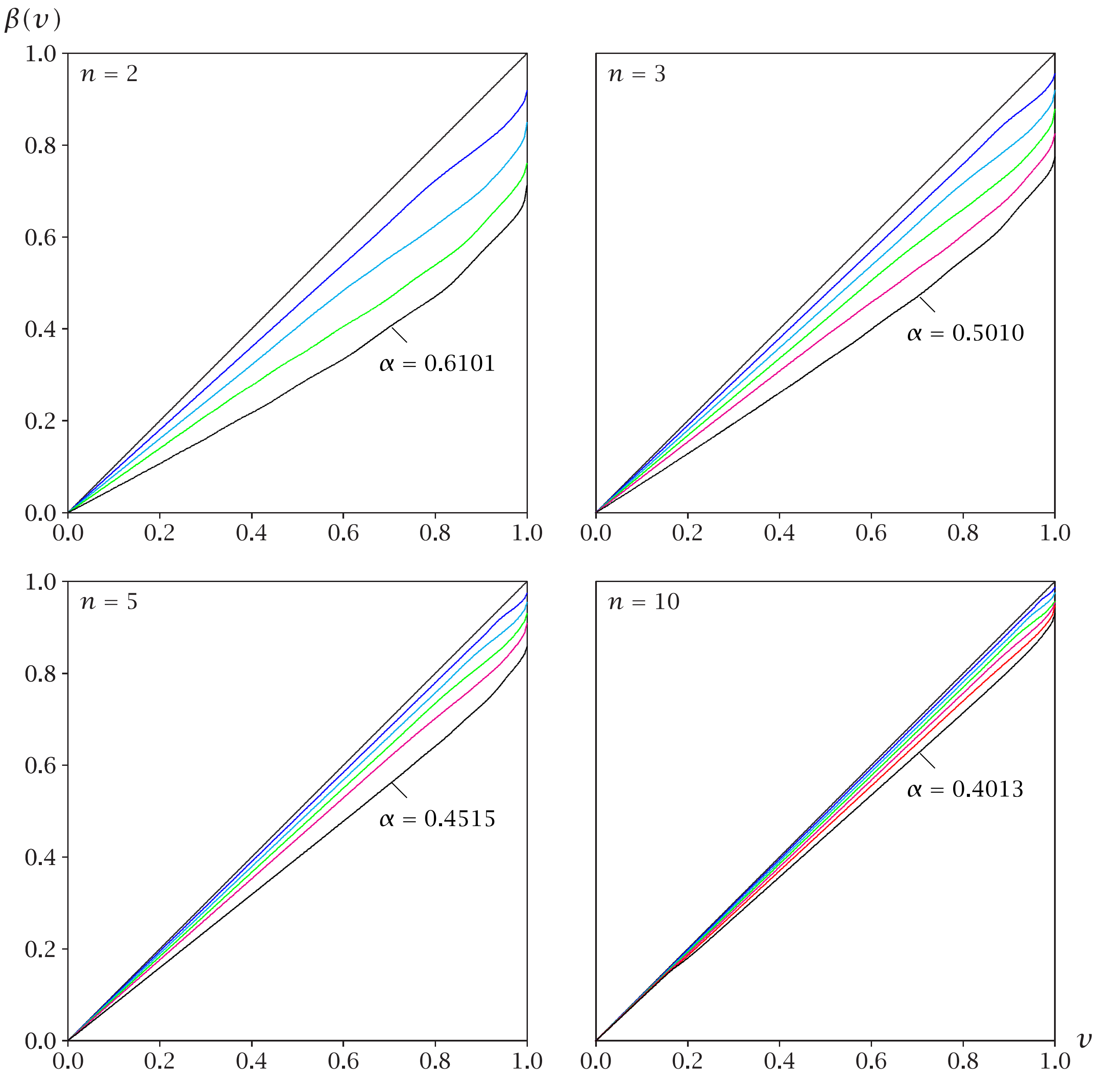

Figure 1: Approximate equilibrium bid functions for the first-price auction with type I and type II corruption, for $n \in \{2, 3, 5, 10\}$ and $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and, for each n, for the smallest α for which we found a solution.

Table 1: Precision of the computations for the first-price auction with both types of corruption.

$n=2$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	32	4.40×10^{-22}	2.09×10^{-12}	2.04×10^{-11}	-1.43×10^{-16}
	0.8	34	2.23×10^{-17}	4.71×10^{-10}	3.94×10^{-9}	$+8.86 \times 10^{-14}$
	0.7	100	9.51×10^{-15}	9.73×10^{-9}	1.06×10^{-7}	$+6.23 \times 10^{-11}$
	0.6101	43	6.17×10^{-12}	2.48×10^{-7}	3.50×10^{-6}	-1.78×10^{-8}
	0.6100	25	9.76×10^{-8}	3.12×10^{-5}	2.96×10^{-4}	-7.41×10^{-6}
	0.6	29	3.04×10^{-8}	1.74×10^{-5}	1.84×10^{-4}	-2.06×10^{-6}
	0.5	61	5.51×10^{-2}	2.34×10^{-2}	6.33×10^{-2}	-1.95×10^{-2}

$n=3$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	46	3.82×10^{-28}	1.95×10^{-15}	1.80×10^{-14}	$+6.49 \times 10^{-17}$
	0.8	31	4.39×10^{-25}	6.61×10^{-14}	3.72×10^{-13}	-1.97×10^{-16}
	0.7	50	9.73×10^{-24}	3.11×10^{-13}	2.77×10^{-12}	-2.28×10^{-16}
	0.6	37	2.15×10^{-24}	1.46×10^{-13}	1.38×10^{-12}	$+2.20 \times 10^{-15}$
	0.5010	50	3.98×10^{-16}	1.99×10^{-9}	2.82×10^{-8}	-1.37×10^{-10}
	0.5009	58	5.73×10^{-9}	7.55×10^{-6}	6.58×10^{-5}	$+3.13 \times 10^{-7}$
	0.5	25	1.00×10^{-1}	3.15×10^{-2}	2.22×10^{-1}	-6.96×10^{-3}

$n=5$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	19	7.18×10^{-7}	8.45×10^{-5}	8.85×10^{-4}	-8.19×10^{-7}
	0.8	99	6.32×10^{-21}	7.93×10^{-12}	1.11×10^{-10}	$+4.07 \times 10^{-13}$
	0.7	98	5.85×10^{-20}	2.41×10^{-11}	2.59×10^{-10}	-2.41×10^{-12}
	0.6	16	8.84×10^{-17}	9.38×10^{-10}	7.90×10^{-9}	-1.15×10^{-10}
	0.5	99	7.64×10^{-19}	8.72×10^{-11}	8.17×10^{-10}	-5.11×10^{-12}
	0.4515	98	2.55×10^{-18}	1.59×10^{-10}	1.91×10^{-9}	-1.93×10^{-11}
	0.4514	18	1.24×10^{-5}	3.51×10^{-4}	4.47×10^{-3}	$+9.85 \times 10^{-6}$
	0.4	13	2.11×10^{-1}	4.59×10^{-2}	2.30×10^{-1}	-1.86×10^{-2}

$n=10$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	19	2.46×10^{-7}	4.95×10^{-5}	3.33×10^{-4}	-1.16×10^{-5}
	0.8	44	9.75×10^{-18}	3.11×10^{-10}	1.74×10^{-9}	-7.96×10^{-11}
	0.7	98	6.07×10^{-18}	2.46×10^{-10}	1.47×10^{-9}	-5.77×10^{-11}
	0.6	100	4.15×10^{-17}	6.43×10^{-10}	3.78×10^{-9}	-1.55×10^{-10}
	0.5	98	1.25×10^{-16}	1.12×10^{-9}	6.38×10^{-9}	-2.97×10^{-10}
	0.4013	37	2.95×10^{-15}	5.42×10^{-9}	3.08×10^{-8}	-1.61×10^{-9}
	0.4012	43	8.74×10^{-10}	2.95×10^{-6}	2.93×10^{-5}	-4.56×10^{-8}
	0.4	100	2.17×10^{-5}	4.65×10^{-4}	2.09×10^{-3}	-2.44×10^{-4}

`results.xls`, containing some statistics and graphs of the computations. Also included are plain text files which are copies of the output produced by the program. The naming conventions of these files are as follows: the file with name `x-0y.txt` contains the computation for $n = \mathsf{x}$ and $\alpha = 0.\mathsf{y}$.

All parameters and choices of methods are specified at the beginning of the source code (Table 2), so you do not need to modify the program itself when making your choices.

Table 2: Section of the source code in which all parameters are declared.

```
// ===============================================================================
// In this section, the parameters of the problem are given.
// You are free to change these parameters.

const byte n = 2;                  // # bidders (greater than or equal to 2)
const double alpha = 0.9;          // the bidder's share (between 0.0 and 1.0)

// ===============================================================================
// Next we define the parameters defining the discrete approximation, the methods
// for find the root and the initial guess, and the stopping conditions.
// These settings should only be changed with caution.

// --- discrete valuation space ---------------------------------------------------

static int g = 200;                     // initial # of points in the space of valuations
const int SubdivisionFactor = 4;        // factor by which valuation space is subdivided
const int NbSubdivisions = 0;           // number of times the space is subdivided

// "delta" for finite difference derivatives
const double diffDelta = 1E-10;         // this is the "delta" we use to compute the
                                        // derivatives
// --- initital bid function -----------------------------------------------------

const int initmethod = 0;               // choice of method for initial guess
                                        // initmethod = 0 : linear guess
                                        // initmethod = 1 : grid search

const double slope = (n-1)/(n-alpha);   // slope of linear initial guess

const int GridSearchSteps = 50;         // parameter for the grid search
                                        // GridSearchSteps = # values that are tried for
                                        // each point in the valuation grid
```

→ *continued on next page*

(Table 2 continued)

```
// --- step size optimization ----------------------------------------------------------

const int taumethod = 1;        // taumethod = 0: brute force
                                // taumethod = 1: golden section search

const int tausteps = 10;        // number of stepsizes we try out in each iteration step
                                // when using brute force
const double tauEps = 1E-2;     // required precision when using golden section search

const double GRANDMAXTAU = 2.0; // maximum step size under all circumstances

// --- root finding --------------------------------------------------------------------

const int rootmethod = 3;       // choice of method for root finding
                                // rootmethod = 0 : steepest descent method
                                // rootmethod = 1 : Gauss-Newton method (inverse Jacobian)
                                // rootmethod = 2 : hybrid method
                                // rootmethod = 3 : Levenberg-Marquardt method

// switching rule for hybrid method
const double switchEps = 1E-12;
const double switchKeep = 5;

// coefficients for Levenberg-Marquardt method
const double initMu = 1E-3;             // initial mu
const double muFactor = 10;             // factor by which mu is multiplied or divided
const double maxMu = 1E+100;            // upper bound for mu
const double minMu = Double.Epsilon;    // lower bound for mu

// --- stopping rules ------------------------------------------------------------------

const double eps = Double.Epsilon;  // stop iterating if SSE < eps
                                    // (Double.Epsilon is machine precision, so this
                                    // effectively turns off this stopping rule; you
                                    // can choose a larger value for eps)
const int keepiter = 20;            // stop iterating if no improvement in so many steps
const int maxiter = 100;            // stop after so many steps in any case

// =====================================================================================
```

References

HEATH, M. T. (2002): *Scientific Computing. An Introductory Survey*. McGraw-Hill, 2nd edn.