Revised Draft: 28 February 1994

Contribution to: H. Amman, D. Kendrick, and J.Rust (eds.),
Handbook of Computational Economics, Volume 1,
Elsevier Science Publishers, 1995.

Chapter I.E.
Nonlinear Pricing and Mechanism Design

Robert Wilson *

In applications of theories of incentives, the information known privately by an economic agent is represented by a point in a Euclidean space. Other agents know the probability distribution of this point, but not its realization, which is called the agent's *type*. For models of this sort, designs of optimal incentive schemes present few difficulties when agents' types are one-dimensional. The computational difficulties are severe, however, when the types are multidimensional. When the types are m-dimensional, the main task is to solve a family of partial differential equations to obtain a map $\mu : \Re^m \to \Re^m$ that provides the Lagrange multipliers for each type's incentive-compatibility constraints. This chapter describes methods for solving simple versions that arise in nonlinear pricing and mechanism design.

1. Mirrlees' Formulation

To illustrate the origin of the central computational problem, we present the formulation of nonlinear pricing introduced by James Mirrlees (1971, 1976). [1] This formulation characterizes the design of a tariff offered by a firm to a customer whose preferences the firm does not know.

Statement of the Nonlinear Pricing Problem

Consider a monopolist seller who charges a tariff $P(q)$ for a bundle q of its products.

* Stanford Business School, Stanford, CA 94305-5015. Ph: 415-723-8620. Fax: 725-7979. Internet: RW@Yen.Stanford.edu.

[1] Mirrlees' initial formulation focused on optimal taxation. Nonlinear versions of Ramsey pricing and taxation are variants of the general principal-agent problem affected by adverse selection and/or moral hazard. A formulation in the context of mechanism design is presented in Wilson (1993b), which includes extensions to cases where agents' types are correlated. For surveys of other applications, see Roger Guesnerie and Jean-Jacques Laffont (1984) and Wilson (1993a, §15).

If wealth effects and risk aversion are absent, then a customer of type t is predicted to respond with the purchase $q(t)$ that maximizes his net benefit $U(q,t) - P(q)$ among the feasible bundles $q \in Q$.[2] Here, the utility function U measures the customer's gross benefit in money terms, depending on both the bundle purchased and the customer's type. The customer knows his type but the seller does not. The seller's objective is to offer the tariff that maximizes its expected revenue.

Assume hereafter that Q is the set of nonnegative bundles in an ℓ-dimensional Euclidean space, and adopt the normalization $U(0,t) = 0$. The set T of possible types is represented similarly as a compact, convex, full-dimensional subset of an m-dimensional Euclidean space, and it has a piecewise-smooth boundary. Assume further that U is a smooth increasing function of both the bundle and the type, and a concave function of the bundle. It is also usual to impose conditions ensuring the existence of a tariff that induces self-selection by the different types of the customer: in the one-dimensional case with a single commodity and a single type parameter (i.e., $\ell = m = 1$), a typical condition requires that U_{qt} is uniformly positive and the probability distribution of the type has an increasing hazard rate; cf. Wilson (1993a, §8).[3]

The important aspect of Mirrlees' formulation of the seller's problem is to construct the solution of a relaxed version in which some of the possibly relevant constraints are omitted. In the single-type case it is known that fairly weak conditions suffice to ensure that the solution of the relaxed problem is the solution of the complete problem; or, the solution can be obtained from a simple modification (called the ironing procedure) described by Mussa and Rosen (1978), Guesnerie and Laffont (1984), and Wilson (1993a, §8). Although comparable sufficiency conditions have not been established for multidimensional formulations, here we present computational methods only for the relaxed problem.[4]

The motivation for the relaxed problem used in Mirrlees' formulation stems from

[2] Formulations that include risk aversion and wealth effects are presented in Mirrlees (1976, 1986), Kevin Roberts (1979), and Wilson (1993), among many others. We focus on a formulation without these effects, which captures the key ingredients of the main computational problem.

[3] In multidimensional versions, the relevant conditions on the utility function are the 'increasing differences' and the 'single crossing' properties; cf. Milgrom and Shannon (1994).

[4] Various sufficiency conditions invoked to enable representation of the customer's optimality condition by the envelope property are described by Mark Armstrong (1992, 1993), Mirrlees (1976, 1986) and Wilson (1993a, §8), though mainly for one-dimensional formulations.

the following considerations. Given the tariff P, the net benefit of type t is

$$W(t) = \max_{q \in Q} U(q,t) - P(q) .$$

Alternatively, given the net benefit $W(t)$, the seller's revenue from type t is

$$R(t) \equiv P(q(t)) = U(q(t),t) - W(t) . \tag{1}$$

In this alternative representation, the optimality of the bundle $q(t)$ is conveyed, in part, by two necessary conditions.

- *Incentive-compatibility* constraint:

$$W_t(t) = U_t(q(t),t) , \tag{2}$$

  if the bundle $q(t)$ is in the interior of Q.

The incentive-compatibility constraint states the envelope property implied by the customer's optimization. Because the type t has m dimensions, each side of the incentive-compatibility constraint is a gradient vector; e.g., $U_t(q,t) \equiv \langle \frac{\partial}{\partial t_i} U(q,t) \rangle_{i=1,\ldots,m}$.

- *Participation* constraint:

$$W(t) \geq 0 . \tag{3}$$

The participation constraint recognizes that the customer retains the option to forego purchases.

Auxiliary Constraints

These constraints are just two of the necessary conditions required for a solution of the customer's optimization problem. To illustrate some of the other conditions that are potentially relevant, we describe three. One is evident if U is a convex function of the type: in this case W must be a convex function, since it is obtained as the pointwise maximum of a family of convex functions indexed by the bundle. Another is that the customer's local second-order necessary condition for a maximum must be satisfied. For instance, suppose there exists a smooth mapping $t(q)$ specifying the type purchasing bundle q. From the customer's first-order necessary condition one infers that the vector of marginal prices at q is $P_q(q) = U_q(q,t(q))$. Then the second-order condition that requires $U_{qq}(q,t) - P_{qq}(q)$ to be negative semi-definite at $q = q(t)$ implies that $U_{qt}(q,t(q)) \cdot t_q(q)$ must be positive semi-definite. In the one-dimensional case, if $U_{qt} > 0$ then this requires that $t(q)$, and therefore also $q(t)$, are nondecreasing functions.

A third is the requirement that $U_t(q(t),t)$ must be integrable to obtain $W(t)$, as described below in (7). For a detailed elaboration of these auxiliary conditions and how the solution of the relaxed problem can be modified to satisfy them in the one-dimensional case, see Guesnerie and Laffont (1984). Here we ignore these auxiliary conditions and develop computational methods only for the relaxed problem.

The Relaxed Problem

Assume for simplicity that the seller's costs are nil, so that its objective is to maximize its expected revenue. Let $f(t)$ be the probability density that the customer's type is t (or the number of customers of that type), defined on the support $T \subset \Re^m$ of possible types. Then the seller's optimization problem can be cast as choosing the two functions $q(t)$ and $W(t)$ to maximize its expected revenue

$$\int_T R(t)\,dt \equiv \int_T [U(q(t),t) - W(t)]f(t)\,dt_1 \cdots dt_m\,,$$

subject to the incentive-compatibility and participation constraints. Note that this formulation converts the seller's optimization from the assignment of a tariff to each bundle, to the choice of an assignment of a bundle and a net benefit to each type.

Necessary Conditions for a Solution

To address this problem, let $\mu(t)$ be an m-dimensional Lagrange multiplier attached to the incentive-compatibility constraint. The Lagrangian form of the objective function is then

$$\int_T \{[U(q(t),t) - W(t)]f(t) + [W_t(t) - U_t(q(t),t)] \cdot \mu(t)\}\; dt_1 \cdots dt_m\,.$$

This objective presents a classical problem in the calculus of variations. On the assumption that W and μ are smooth functions, three necessary conditions for an optimum are the following:[5]

- Allocative Optimality of the assignment of bundles:

$$U_q(q(t),t)f(t) - U_{qt}(q(t),t) \cdot \mu(t) \leq 0\,, \tag{4}$$

 and this inequality is complementary to the feasibility constraint $q(t) \geq 0$.

[5] Two inequalities $a \leq 0$ and $b \geq 0$ are *complementary* if their inner product is nil: $a \cdot b = 0$.

- Welfare Optimality of the assignment of net benefits:

$$-f(t) - \sum_{i=1}^{m} \frac{\partial \mu_i}{\partial t_i}(t) \leq 0\,, \tag{5}$$

and this inequality is complementary to the participation constraint $W(t) \geq 0$ on the interior of the type domain. This is an Euler condition; the sum is called the *divergence* of μ.

- Optimality on the boundary: if $\nu(t)$ is the outward-pointing normal vector at a point $t \in \partial T$ on the boundary of the domain of types, then

$$\nu(t) \cdot \mu(t) \leq 0\,, \tag{6}$$

and this inequality is complementary to the participation constraint $W(t) \geq 0$ on the boundary of the type domain. This is a transversality condition.

These conditions indicate that the key step in obtaining a solution is to construct the Lagrange multiplier μ. [6] Once this multiplier has been obtained, the optimal assignment of types to bundles is obtained by solving the ordinary equation (4). Moreover, at the bundle $q = q(t)$ the vector $p(q) \equiv P_q(q)$ of marginal prices must be $p(q) = U_q(q,t)$. The tariff $P(q)$ is therefore obtained by integrating these marginal prices, using the participation constraint (where it binds) to determine the constant of integration.

The main computational task, therefore, is to construct the multiplier μ by solving the welfare-optimality condition (5) and the transversality condition (6), interpreted as equalities on T and ∂T. It is important to realize, however, that if $m > 1$ then the single partial differential equation (5) and the boundary condition (6) are insufficient to determine the m components of the multiplier. For this one needs $m - 1$ additional conditions derived from the requirement that the marginal prices must be integrable to obtain the tariff. An equivalent requirement is that the gradient W_t of the net benefit must be an integrable function of the type (Mirrlees, 1986, p. 1241). Using the incentive-compatibility condition, this requirement provides a fourth condition.

- *Integrability* condition: the vector field

$$U_t(q(t), t) \quad \text{is integrable.} \tag{7}$$

[6] The alternative eliminates $\mu(t)$ by substituting (4) into (5) and (6) to solve for $q(t)$. If U_q is linear then this has the same difficulty as solving for μ, and otherwise it involves nonlinear partial differential equations.

With sufficient smoothness assumptions, this condition is equivalent to the requirement that the Hessian matrix $W_{tt}(t)$ is symmetric. Moreover, it is sufficient that only $m-1$ of the symmetry conditions are satisfied:

$$\frac{\partial^2 W}{\partial t_i t_{i+1}}(t) = \frac{\partial^2 W}{\partial t_{i+1} t_i}(t), \qquad i = 1, \ldots, m-1 .$$

Thus the integrability condition imposes $m-1$ additional partial differential equations that with (5) and (6) ordinarily suffice to determine the multiplier.

When $m = 1$, the integrability condition is vacuous and the welfare-optimality and transversality conditions have the trivial solution $\mu(t) = 1 - F(t)$ independently of U, where F is the distribution function for the density f.[7] To illustrate the form of the integrability condition when $\ell = m \geq 2$, suppose

$$U(q,t) = q^T \cdot [A \cdot t - \frac{1}{2} B \cdot q ,$$

$$\text{so that} \qquad U_q(q,t) = A \cdot t - B \cdot q ,$$

where A and B are matrices, and B is symmetric and positive definite.[8] Then the vector field $W_t(t) = q(t)^T \cdot A$ must be integrable, where $q(t) = B^{-1} \cdot A \cdot [t - \mu(t)]$. The corresponding Hessian matrix is $W_{tt}(t) = C \cdot [I - \mu_t(t)]$, where the matrix $C \equiv A^T \cdot B^{-1} \cdot A$ is symmetric and positive definite. Consequently, integrability requires that $C \cdot \mu_t(t)$ is a symmetric matrix. When $m = 2$, therefore, the condition that ensures integrability is an equality between the off-diagonal elements of this matrix:

$$c_{11} \frac{\partial \mu_1}{\partial t_2} + c_{12} \frac{\partial \mu_2}{\partial t_2} = c_{21} \frac{\partial \mu_1}{\partial t_1} + c_{22} \frac{\partial \mu_2}{\partial t_1} ,$$

where $c_{12} = c_{21}$ by construction.

The interplay among the four conditions when $m > 1$ is illustrated in the following example, which allows a closed-form solution.

---

[7] When the hazard rate of F is not increasing the customer's second-order necessary condition for an optimum need not be satisfied. In this case, it may be necessary to apply the ironing procedure to modify the bundle assignment so that it is nondecreasing; cf. Guesnerie and Laffont (1984) and Wilson (1993a, §8).

[8] The increasing-differences property requires essentially that A has nonnegative elements, and the super-modularity version of the single-crossing property requires that the off-diagonal elements of B are nonpositive; cf. Milgrom and Shannon (1984). Invoking super-modularity requires that the type space is a complete lattice, such as an m-dimensional cube.

Example 1: For this example, $\ell = m$ and $A = B = I$ so

$$U(q,t) = \sum_{i=1}^{m} \left(t_i q_i - \frac{1}{2} q_i^2 \right) ,$$

and $f(t) = 1$ on the domain

$$T = \{ t \mid t_i \geq 0 \ \& \ \sum_i t_i^2 \leq r^2 \} ,$$

which is the positive orthant of the ball with radius r. To derive the solution analytically, we guess that $W(t) = w(y(t))$, where w is a univariate function and $y(t) = \sum_i t_i^2$. If this specification allows a choice of w satisfying the other conditions, then the integrability condition (7) is satisfied automatically. From the condition (4) for an optimal assignment we obtain $q(t) = t - \mu(t)$ and from the incentive-compatibility condition (2) we obtain $W_t(t) = q(t)$, so the specification implies that

$$w'(y(t))2t = t - \mu(t) .$$

Differentiating this relationship and then summing yields

$$w''(y(t)) \sum_i [2t_i]^2 + w'(y(t))2m = \sum_i \left[1 - \frac{\partial \mu_i}{\partial t_i}(t) \right] .$$

Invoking the welfare-optimality condition (5) yields

$$w''(y)4y + w'(y)2m = m + 1 ,$$

which is a differential equation for w' as a function of y. On the upper boundary where $y(t) = r^2$, the outward-pointing normal is proportional to $\nu(t) = 2t$, so the transversality condition (6) imposes the boundary condition $2t \cdot \mu(t) = 0$, or equivalently

$$\sum_i 2t_i[t_i - w'(r^2)2t_i] = 0 ,$$

which requires $w'(r^2) = 1/2$. The unique solution of the differential equation that satisfies this boundary condition is

$$w'(y) = \frac{1}{2} \left(1 + \frac{1}{m} \left[1 - \frac{r^m}{y^{m/2}} \right] \right) .$$

Consequently, the multiplier is

$$\mu(t) = \frac{1}{m}\left[\left(\frac{r}{\sqrt{y(t)}}\right)^m - 1\right] t .$$

Fortuitously, this multiplier also satisfies the transversality condition $\mu_i(t) = 0$ on the lower boundary where $t_i = 0$. Therefore, it satisfies all of the conditions required.

Similarly, when the density is exponential, say $f(t) = e^{-y(t)}$ on the domain $\{t \mid y(t) \leq r^2\}$, one obtains

$$w'' \sum_i y_i^2 + w' \sum_i [-y_i^2 + y_{ii}] = m + 1 - \sum_i y_i t_i$$

$$\text{and} \qquad \mu_i(t) = f(t)[t_i - w'(y(t))y_i(t)] ,$$

where $y_i \equiv \frac{\partial}{\partial t_i} y$, provided this yields w' as a function solely of y. For instance, if $y(t) = \frac{1}{2}\sum_i t_i^2$ so that the density is Normal with $r = \infty$, and m is even, then

$$\mu(t) = \frac{1}{2} f(t) \left(\sum_{k=1}^{m/2} a_k / y^k\right) t ,$$

where $a_1 = 1$ and $a_{k+1} = [m/2 - k]a_k$. Caution is required, however; e.g., if $y(t) = \sum_i t_i$ then the resulting multiplier cannot satisfy the transversality conditions, which stems from the feature that the resulting optimal bundles lie along a 1-dimensional ray. Hereafter we assume that the range of the bundles is full-dimensional to exclude such complications.

Example 1 is one of a rare collection allowing closed-form solutions; others that also rely on exploiting radial symmetries are described in Armstrong (1992, 1993) and Wilson (1993a, §13.5; 1993b, p. 145 ff.).[9] The following sections therefore describe numerical methods for solving the differential equations (5) and (7) with the boundary condition (6).[10] We concentrate on the case $\ell = m = 2$. As we shall see, this case with two products and two type parameters shows already that the problem poses computational difficulties.

[9] In such examples the optimal tariff depends only on a one-dimensional measure of the aggregate 'size' of the bundle purchased. The analysis of Example 1 in Wilson (1993a, p. 340) has an error in the leading coefficient of μ_i.

[10] One cannot impose (5) and (6) as equalities throughout T because the Divergence Theorem would then imply $\int_T f(t)\,dt = 0$. Typically, they are strict inequalities for 'small' types for whom $W(t) = 0$; cf. Armstrong (1992, 1993).

For later reference we mention also a second, closely related example for which the method used for Example 1 is evidently insufficient.

Standard Test Problem: This example is the same as Example 1 with $\ell = m$ and $f(t) = 1$, but the support T is the unit square. In spite of its simplicity, this example does not seem to be solvable in closed form. In addition, it is peculiar in that it has a continuum of other 'solutions' whose non-optimality is revealed only by the fact that they are discontinuous. For instance, one such 'solution' is

$$\mu_i(t) = \begin{cases} m[1 - t_i] & \text{if } t_i \geq \max_{j \neq i} t_j \,, \\ t_i & \text{if } t_i < \max_{j \neq i} t_j \,, \end{cases}$$

which for $m = 2$ is discontinuous along the diagonal of the unit square. These extraneous solutions indicate that caution is required in using algorithms that rely on discrete approximations. Such algorithms usually provide no assurance that the differentiable solution, which is the only optimal one, is the one approximated by numerical calculations.

2. Computational Methods

In this section we describe numerical algorithms for solving Mirrlees' relaxed problem. Several are described only for the standard test problem.

Direct Optimization

We mention first the computational method used by Armstrong (1992). This method solves directly a discrete version of the seller's optimization problem. Although simplest in design, it is uneconomical in terms of storage and speed.

The set T of possible types is represented by a finite set $\hat{T}$ comprising a discrete grid of points with mesh size δ. The seller's problem is then cast as the nonlinear constrained optimization problem in which the objective is to choose the nonnegative variables $\langle q(t), W(t) \rangle_{t \in \hat{T}}$ to maximize the expected profit

$$\sum_{t \in \hat{T}} [U(q(t), t) - W(t)] f(t) \delta^m$$

subject to the discretized incentive-compatibility constraints

$$W(t + \delta e_i) - W(t) = \frac{\partial U}{\partial t_i}(q(t), t) \delta \,,$$

for each point $t \in \hat{T}$ and each $i \leq m$, where e_i is the i-th unit vector. If T is the unit cube and $\delta = 1/n$ then this formulation requires $[m+1]n^m$ variables and mn^m (generally nonlinear) constraints, in addition to the nonnegativity constraints. Thus when $m = 2$ the coarse grid with $n = 20$ requires 1200 variables and 800 constraints. Nonlinear optimization problems of this size are feasible but storage and time requirements are substantial. Armstrong (1992) reports results for the standard test problem, for which the incentive-compatibility constraints are linear and symmetry considerations allow half as many variables. This method seems to have little prospect of being feasible and accurate when the number m of type dimensions exceeds two.

Approximation via Fourier Series

For problems in which the utility function U has a simple analytical form it may be feasible to derive a Fourier (or other finite-element) approximation. The technique relies on the derivation of an auxiliary second-order partial differential equation for each multiplier, from which a class of solutions represented as Fourier series can be constructed.

We illustrate using the standard test problem. For this problem the bundle purchased by type t is $q(t) = t - \mu(t)$ and the incentive-compatibility constraint requires that $W_t(t) = q(t)$. Consequently, the integrability condition that the Hessian matrix $W_{tt}(t)$ is symmetric requires that the Jacobian matrix $\mu_t(t)$ is symmetric. Thus, a solution requires that

$$\frac{\partial \mu_1}{\partial t_2}(t) - \frac{\partial \mu_2}{\partial t_1}(t) = 0$$

in addition to the welfare-optimality condition

$$\frac{\partial \mu_1}{\partial t_1}(t) + \frac{\partial \mu_2}{\partial t_2}(t) + f(t) = 0 .$$

Differentiating these two conditions with respect to t_2 and t_1 respectively yields a single second-order equation for $\mu_1(t)$:

$$\frac{\partial^2 \mu_1}{\partial t_1^2}(t) + \frac{\partial^2 \mu_1}{\partial t_2^2}(t) + \frac{\partial f}{\partial t_1}(t) = 0 ,$$

and analogously for μ_2. Thus one obtains a classical Poisson equation for each multiplier separately. The problem is not of the standard Dirichlet or von Neumann form, however, because the boundary condition where $t_2 \in \{0, 1\}$ depends on the solution of the analogous problem for the other multiplier.

Similarly, if U_q is linear as in (8) and $c_{ii} = 1$ and $c_{12} = \gamma$ then one obtains the elliptic equation

$$\frac{\partial^2 \mu_1}{\partial t_1^2}(t) + \frac{\partial^2 \mu_1}{\partial t_2^2}(t) - 2\gamma \frac{\partial^2 \mu_1}{\partial t_1 \partial t_2}(t) + \frac{\partial f}{\partial t_1}(t) - \gamma \frac{\partial f}{\partial t_2}(t) = 0 .$$

Similar equations arise in higher-dimensional versions with constant coefficients.

In the standard test problem, the density is $f(t) = 1$ on the unit square, and therefore $\frac{\partial}{\partial t_i} f = 0$, which yields a classical Laplace equation. For the transversality condition that imposes the boundary condition $\mu_1(t) = 0$ for $t_1 \in \{0,1\}$ and $t_2 > 0$, Fourier representations of the multiplicatively separable solutions of the Laplace equation take the form (cf. Milne, 1970, §10; and Derrick and Grossman, 1987, §10):

$$\mu_1(t) = \sum_{k=1}^{n} a_k \sin(k\pi t_1) \cosh(k\pi[1 - t_2]) ,$$

where the constants a_k are arbitrary. Moreover, if μ_1 has this form then the welfare-optimality condition implies that

$$\mu_2(t) = 1 - t_2 + \sum_{k=1}^{n} a_k \cos(k\pi t_1) \sinh(k\pi[1 - t_2]) ,$$

where we have fixed the constant of integration to satisfy the transversality condition at $t_2 = 1$. The transversality condition on the remaining portion of the boundary where $t_2 = 0$ imposes the boundary condition $\mu_2(t) = 0$ there, at least for those types for which t_1 is large enough that the net benefit $W(t)$ is positive. This yields a set of linear equations of the form

$$\sum_{k=1}^{n} b_k \cos(k\pi t_1) + 1 = 0 ,$$

where $b_k \equiv a_k \sinh(k\pi)$, to be solved for a set of n values of t_1 to determine the coefficients b_k. In fact, if the chosen points are $t_1(k) = 1 - [k-1]\delta$ where the mesh size is $\delta = 1/n$, then $b_k = 2$ except that $b_n = 1$.

In sum, the Fourier method consists of the following steps. First one derives a Poisson equation for one of the multipliers, say μ_1. This equation is derived by differentiating the welfare-optimality condition and the integrability condition, and then solving the resulting equations to eliminate the other multiplier, if this is possible in closed form. From the Poisson equation one establishes a family of Fourier solutions satisfying the boundary conditions for μ_1 on the segments of the boundary where $t_1 = 0$

or $t_1 = 1$. From the welfare-optimality condition, interpreted as an ordinary first-order differential equation for μ_2, one then obtains a corresponding family of Fourier solutions for μ_2. The last step is to determine the coefficients from the boundary conditions for μ_2 on the remaining segments of the boundary. This method for the two-dimensional case indicates the outline of a method for higher-dimensional cases, but no examples have been solved.

Introduction to Finite-Difference Methods

Algorithms based on finite-difference approximations encounter fundamental difficulties. To motivate the circuitous approach taken in the design of the algorithms described later, we first describe the source of these difficulties. We use $\mu_i[j,k]$ to indicate the value of $\mu_i(t)/\delta$ at a grid point $t = (t_1, t_2)$ where $t_i = j\delta$ and $t_2 = k\delta$ for the standard test problem with mesh size δ.

With this notation, a naive approach to representing the welfare-optimality and integrability conditions in terms of finite differences produces the two equations

$$\begin{aligned} \big(\mu_1[j+1,k] - \mu_1[j,k]\big) + \big(\mu_2[j,k+1] - \mu_2[j,k]\big) + f(j\delta, k\delta) &= 0\,, \\ \big(\mu_1[j,k+1] - \mu_1[j,k]\big) - \big(\mu_2[j+1,k] - \mu_2[j,k]\big) &= 0\,, \end{aligned} \tag{9}$$

at each interior grid point, and analogous equations on the boundary (excluding the origin). In fact, however, this approach is doomed to failure: such equations are invariably both singular (redundant columns in the associated matrix) and inconsistent (redundant rows with incompatible constant terms)!

These deficiencies stem from an important economic consideration. Although the welfare-optimality condition is properly formulated in terms of forward differences, the integrability condition must be formulated in terms of backward differences. This reflects the orientation of the incentive-compatibility constraint, which in the discrete version is binding for lower types.

Three schemes are described below. Each takes account in a different way of the opposite directional orientations of the two conditions. The first two rely on a relaxation algorithm, and later we describe an alternative scheme that relies on a direct algorithm. [11]

Relaxation Combined with Newton's Method

The Newton-Relaxation Algorithm takes the conservative view: all directions are included in the construction of the values at one grid point from the values at its neighbors.

[11] See Golub and Ortega (1992, §9) for expositions of these standard algorithms.

This tactic is implemented by invoking two refinements in the formulation.

The first refinement introduces the analogs of the equations (9) in which the stencil is rotated. Solving (9) yields

$$\mu_1[j,k] = \frac{1}{2}\left(f(j\delta, k\delta)\delta + \mu_1[j+1,k] + \mu_1[j,k+1] + \mu_2[j,k+1] - \mu_2[j+1,k]\right) ,$$

and analogously for $\mu_2[j,k]$. This representation is the basis for a recursive formula that relies on a three-point stencil in which the values $\mu_1[j,k]$ and $\mu_2[j,k]$ are constructed from the values at the two adjacent points *higher* in the grid. Rotating this stencil through 90, 180, and 270 degrees, and then adding the four formulas produces the recursion

$$\mu_1[j,k] = \frac{1}{4}\left(\mu_1[j+1,k] + \mu_1[j,k+1] + \mu_1[j-1,k] + \mu_1[j,k-1]\right)$$

for the standard test problem. Notice that adding the four rotations of the basic stencil eliminates the dependence on adjacent values of the other multiplier. Moreover, this recursion is precisely the one obtained from the natural five-point stencil derived from the Laplace equation. The recursion is symmetric, produces a nonsingular system of equations, and yields a relaxation algorithm that is unconditionally stable.[12] In more general problems, one wants assurance that inclusion of the rotations of the basic stencil will suffice to assure that it is the differentiable solution that is approximated by the calculations. No general theoretical justification for this conclusion seems to be known, however, and therefore one must rely pragmatically on the encouraging evidence from computational experience.

The second refinement adapts the transversality condition to obtain a formulation that relies on only the one multiplier μ_1 occurring in the recursion used for the relaxation algorithm. No modification is required on the two sides of the unit square where $t_1 \in 0,1$: there the boundary condition is $\mu_1(t) = 0$. To obtain a boundary condition on the side where $t_2 = 1$, note that the transversality condition requires $\mu_2(t) = 0$ and therefore $\frac{\partial}{\partial t_1}\mu_2 = 0$. In combination with the integrability condition this implies that the solution must satisfy the *differential constraint*

$$\frac{\partial \mu_1}{\partial t_2}(t) = 0$$

[12] The latter property is well known for elliptic equations; cf. Golub and Ortega (1992). Milne (1970, §10) uses the rotations based on multiples of 45 degrees to obtain a recursion based on a nine-point stencil for which the error from the discrete approximation is smaller by an order of magnitude. In the sequel we omit this refinement as well as others, such as implicit methods, successive over relaxation, and alternating direction methods; cf. Golub and Ortega (1992, §9).

on this segment of the boundary. This is a boundary condition of von Neumann form. The corresponding discrete version imposes the constraint

$$\mu_1[j,n] = \mu_1[j,n-1] .$$

In principle, a similar condition also applies on the opposite side where $t_2 = 0$. However, computational experience shows that it is practically impossible to enforce a differential constraint there with any reasonable mesh size. The reason for the disparity between the two sides where $t_2 \in \{0,1\}$ is evident from solved examples: near the upper boundary the differential constraint is nearly satisfied, whereas near the lower boundary it is far from satisfied. Another relevant consideration is that on the lower boundary the differential condition need not apply for small values of t_1 for which $W(t) = 0$, whereas the monotonicity assumptions imposed on U ensure that it applies uniformly on the upper boundary. Therefore, we enforce the differential constraint only on the upper boundary and seek an alternative condition on the lower boundary.

To obtain a useful boundary condition on the side where $t_2 = 0$, we integrate the welfare-optimality condition over the interval from $t_2 = 0$ to $t_2 = 1$ for each fixed t_1. Taking account of the welfare-optimality constraint yields

$$\phi(t_1) \equiv \int_0^1 \frac{\partial \mu_2}{\partial t_2}(t)\, dt_2 = -\int_0^1 \left(\frac{\partial \mu_1}{\partial t_1}(t) + f(t)\right) dt_2 .$$

Further, the transversality condition requires $\mu_2 = 0$ at both extremes, so we obtain the *integral constraint* that requires

$$\phi(t_1) = 0$$

at each $t_1 > 0$. The corresponding discrete form of the integral value is

$$\phi[j] = -\sum_{k=0}^{n-1} \left(\left(\mu_1[j+1,k] - \mu_1[j,k]\right) + f(j\delta, k\delta)\right) \delta ,$$

In the algorithm, Newton's method is used to improve iteratively the estimates of the $n-1$ values $\langle \mu_1[j,0] \rangle_{0<j<n}$ on the lower boundary until the corresponding integral conditions $\phi[j] = 0$ are satisfied approximately.

Summary of the Newton-Relaxation Algorithm

These ingredients provide the following summary statement of the relaxation algorithm combined with Newton's method. We are given a *symmetric* stencil specifying a discrete recursion amenable to implementation as a relaxation algorithm. Typically this recursion is obtained by summation of the recursions derived from rotations of a basic asymmetric stencil. For the standard test problem, this equation is linear and involves only the one multiplier μ_1, but more generally it may be nonlinear and involve all the multipliers, as we illustrate later.

The only impediment to straightforward calculation of a solution by successive relaxation, therefore, is the unusual set of boundary conditions. Typically these are Dirichlet conditions on a portion of the boundary (where $t_1 \in \{0,1\}$), but on other portions (where $t_2 \in \{0,1\}$) they depend on the other multiplier. To eliminate this dependence, they are replaced by a differential condition (where $t_2 = 1$) and an integral condition (where $t_2 = 1$). On these portions the boundary values are estimated by iterative improvement from an initial guess; in particular, Newton's method is used to improve the approximation of the integral condition.

We use the standard test problem to illustrate. The algorithm starts with an initial approximation of the boundary values $\mu_1(t)$ for $t_2 \in \{0,1\}$ and $t_1 > 0$. This approximation is then improved on the upper boundary by repeatedly applying the differential condition; and on the lower boundary, by repeatedly applying Newton's method to the integral constraints. Between these improvements of the boundary values, the values at interior grid points are obtained by successive relaxation; that is, the symmetric recursion is applied repeatedly.

To initiate the algorithm, we specify an initial guess $\mu_1^0[j,k]$ of the values of $\mu_1(t)$ at all grid points $(j\delta, k\delta) \in \hat{T}$ of the grid, requiring only that $\mu_1^0[0,k] = 0$ on the left side and $\mu_1^0[n,k] = 0$ on the right side of the boundary, for $0 \leq k \leq n$, where the mesh size is $\delta = 1/n$. The subsequent steps of the algorithm alternate between two phases.

Phase 1: Construction of the Interior Values. In this phase the boundary values remain fixed while the symmetric recursion is applied repeatedly. Thus, using an explicit form of relaxation, the new values in iteration $r+1$ are obtained from the previous values by the recursion

$$\mu_1^{r+1}[j,k] = \frac{1}{4}\left(\mu_1^r[j+1,k] + \mu_1^r[j,k+1] + \mu_1^r[j-1,k] + \mu_1^r[j,k-1]\right) ,$$

for $0 < j,k < n$, as derived previously for the symmetric five-point stencil.[13] These iterations are repeated until a test of convergence is satisfied.

[13] This recursion can be improved in accuracy and/or convergence rate by using Milne's nine-point stencil and/or an implicit form, possibly augmented by over relaxation.

Phase 2: Improvement of the Boundary Values. To improve the estimates of the values on the upper and lower boundaries, we proceed as follows.

- On the upper boundary where $t_2 = 1$:

$$\mu_1^{r+1}[j, n] = \mu_1^r[j, n-1],$$

  which enforces the differential constraint $\frac{\partial}{\partial t_2}\mu_1 = 0$ there.

- On the lower boundary where $t_2 = 0$, Newton's method is employed to improve the boundary values to meet the integral constraint:

$$\mu_1^{r+1}[\cdot, 0] = \mu_1^r[\cdot, 0] - \theta J^{-1} \cdot \phi^r,$$

  where $\mu_1^r[\cdot, 0]$ indicates the vector $(\mu_1^r[j, 0])_{0<j<n}$ and similarly $\phi^r[j]$ is the current value of the j-th integral in iteration r. Also, $\theta \in (0,1)$ is a parameter fixing the step size and J is an approximation of the Jacobian matrix of ϕ with respect to $\mu_1[\cdot, 0]$.

After a single iteration of phase 2, one returns to phase 1 to adjust the interior values to conform to the revised boundary values.

Computational experience shows that in practice it is sufficient to use a coarse approximation of the Jacobian matrix. A typical column of this matrix can be constructed by calculating the difference between the integral values obtained from Phase 1 and the integral values obtained when one boundary value (say $\mu_1[n/2, 0]$) is perturbed. Figure 1 shows this difference for the standard test problem. It is evident in the figure that the main effects are a positive increment in its own integral value and a negative increment of equal magnitude in the adjacent integral value. Consequently, it suffices to use a scaled version of the Jacobian matrix for which the diagonal elements are 1, the next-higher diagonal elements are -1, and other elements are zero. This implies that the inverse of the scaled Jacobian, J^{-1}, has elements that are 1 on and above the diagonal, and zero below. This approximation works well even for problems that differ substantially from the standard test problem, including the case (8) in which U_q is linear. A typical value of the step size that works well is $\theta = 0.2$.

The Program nra2L and a Numerical Example

The Appendix includes the APL program *nra2L* that implements this algorithm for the 2-dimensional linear version (8) in which the matrix C (defined previously) conveys the relevant information about the two coefficient matrices A and B. This program assumes that the matrices A and B are symmetric with diagonal elements 1 and off-diagonal elements a and b. Using this algorithm for the case $a = b = 1/2$ and grid size $\delta = 1/40$, one obtains the approximation of the multiplier $\mu_1(t)$ shown in Figure 2 for several values of t_2.

For small values of t_2 the multiplier is large and declines steeply as t_1 increases. For larger values of t_2 the multiplier is smaller, and first *increases* before declining as t_1 increases. The feature that $\frac{\partial}{\partial t_2}\mu_1(t) \approx 0$ when $t_2 \approx 1$ is evident in the figure: for large values of t_2 the curves are close together, and indeed the curve for $t_2 = 0.9$ is virtually indistinguishable from the curve along the upper boundary where $t_2 = 1$.

Because the optimal bundle requires $q_1(t) = \max\{0, t_1 - \mu_1(t)\}$ the customer's purchase of commodity 1 is zero where $\mu_1(t) \geq t_1$. In the figure, therefore, only the region where $\mu_1(t) \leq t_1$ is relevant to the calculation of the optimal assignment and the marginal prices. This eases somewhat the error produced by not enforcing the differential condition on the lower boundary where $t_2 = 0$; however, one must expect that the effects of this error propagate to some extent throughout the type space.

Figure 3 shows the resulting schedule $p_1(q)$ of marginal prices obtained via polynomial interpolation. As with the multiplier, the marginal price $p_1(q)$ first increases and then decreases as q_1 increases. The resulting bundles have the spectacular property ('pure bundling') that each type purchases either both products or neither, which Armstrong (1992, 1993) has shown to be true for a wide class of cases.

It is useful to note the consequences of modifying this example so that $U_q(q,t) = A[t + \epsilon] - Bq$, where ϵ is a positive parameter. In this case, the multipliers remain unchanged but the construction of the bundle $q(t)$ differs significantly. For each sufficiently small value t_2 of the second type, there are now *two* roots of the equation $t_1 + \epsilon_1 = \mu_1(t_1, t_2)$ that defines the lower bound $t_1^*(t_2)$ of the values of the first type t_1 for which $q_1(t_1, t_2) > \epsilon$. In fact, it is the *larger* of these two roots that is the correct one, and the smaller root is extraneous. An additional feature is that the bundle is a discontinuous function of the type, since $q_1(t)$ jumps from zero to ϵ at $\langle t_1^*(t_2), t_2\rangle$. This discontinuity reflects the necessity of including a fixed fee in the tariff.

The Pure Relaxation Algorithm

The pure relaxation algorithm takes a bolder approach, relying on the presumption that formulation of the integrability condition in terms of backward derivatives suffices. It enables a much simpler algorithm, but caution is advised because it has not been tested on enough examples to ensure that this presumption is always justified. As before, we illustrate using the standard test problem.

Using backward derivatives for the integrability condition, the discrete formulation of the welfare-optimality and integrability conditions specifies the two equations

$$\big(\mu_1[j+1,k]-\mu_1[j,k]\big)+\big(\mu_2[j,k+1]-\mu_2[j,k]\big)+f(j\delta,k\delta)=0\,,$$
$$\big(\mu_1[j,k-1]-\mu_1[j,k]\big)-\big(\mu_2[j-1,k]-\mu_2[j,k]\big)=0\,,$$

which yields the recursion

$$\mu_1[j,k]=\frac{1}{2}\left(f(j\delta,k\delta)\delta+\mu_1[j+1,k]+\mu_1[j,k-1]+\mu_2[j,k+1]-\mu_2[j-1,k]\right)\,,$$

and similarly for $\mu_2[j,k]$, based on the symmetric five-point stencil. Phase 1 remains unchanged except that this recursion is used for successive relaxation of *both* multipliers at the interior grid points. In Phase 2 the differential condition remains unchanged too: as before, it is derived solely from the integrability condition and the corresponding transversality condition on each of the two upper boundaries. The significant difference is that now the boundary condition on each of the two lower boundaries can be derived from the welfare-optimality condition and the transversality condition. For instance, on the lower boundary where $t_2=0$ the boundary condition for $\mu_1(t_1,0)$ is

$$\frac{\partial\mu_1}{\partial t_2}(t_1,0)+f(t_1,0)=0\,,$$

which is just the welfare-optimality condition when the transversality condition $\mu_2(t_1,0)=0$ is invoked to get $\frac{\partial}{\partial t_1}\mu_2(t_1,0)=0$. As mentioned previously, however, when t_1 is small this last equality is difficult to enforce with any reasonable grid size; consequently, in practice it is better to use the full form of the corresponding discrete formulation:

$$\mu_1[j-1,0]=\mu_1[j,0]+\mu_2[j,1]+f(j\delta,0\delta)\delta\,,$$

which can be solved recursively starting from $\mu_1[n,0]=0$, as required by the transversality condition on the boundary where $t_1=1$.

This algorithm is implemented in the APL program *pra2L* included in the Appendix. It is written for the general linear form (8) and the type density f is arbitrary; also, it allows that the grid size can differ along the two dimensions of the type domain.

Other Boundary Shapes

As one can see from the contrast between Example 1 and the standard test problem, which differ only in the shape of the upper boundary, the geometry of the type domain is a critical determinant of the multiplier. When the domain is not square the transversality condition involves both multipliers along each boundary segment that is not horizontal or vertical. Such cases require minor modifications of the differential condition and the integral condition.

We illustrate with Example 1. The upper boundary is a segment of a circle and therefore the transversality condition requires that

$$t_1\mu_1(t) + t_2\mu_2(t) = 0$$

along this segment. To obtain the integral condition, therefore, one substitutes the integral formula for $\mu_2(t)$ derived from the welfare-optimality condition to obtain:

$$t_1\mu_1(t) - t_2 \int_0^{t_2} \left(f(t_1, \tau) + \frac{\partial \mu_1}{\partial t_1}(t_1, \tau) \right) d\tau = 0 .$$

The corresponding discrete version is

$$j\mu_1[j,k] - k \sum_{s \leq k} \left(f(j\delta, s\delta) + \mu_1[j+1, s] - \mu_1[j, s] \right) = 0 ,$$

where $t_1 = j\delta$ and $t_2 = k\delta$ on the discrete upper boundary that approximates the actual boundary — or one can use an appropriate interpolation for points on the actual upper boundary.

Similarly, to identify the differential condition for the multiplier μ_1 along this segment, one substitutes the integrability condition $\frac{\partial}{\partial t_2}\mu_1 = \frac{\partial}{\partial t_1}\mu_2$ into the derivative of the transversality condition with respect to t_1 to obtain the formula

$$\mu_1(t) + t_1 \frac{\partial \mu_1}{\partial t_1}(t) + t_2 \frac{\partial \mu_1}{\partial t_2}(t) = 0 .$$

The corresponding discrete version is

$$\mu_1^{r+1}[j,k] = \frac{1}{1+j+k} \left(j\mu_1^r[j-1,k] + k\mu_2^r[j,k-1] \right) .$$

Higher Dimensions

Phase 1 presents no intrinsic difficulties in higher dimensions. For instance, for the analog of the standard test problem when the dimension m is arbitrary, the equation for μ_1 is the general Laplacian

$$\sum_{i=1}^{m} \frac{\partial^2 \mu_1}{\partial t_i^2}(t) = 0 .$$

The analog of the basic recursion used in the relaxation algorithm is therefore the stencil with $2m + 1$ points that constructs $\mu_1(t)$ at the grid point t as the average of the $2m$ values at the adjacent grid points in each of the positive and negative orthogonal directions. In general, as in this example, the stencil is obtained by summing over the stencils obtained from the possible orthogonal rotations of a basic stencil, but if an explicit second-order equation can be derived for each multiplier separately then one can use its natural stencil directly.[14]

In Phase 2, the differential condition is essentially unchanged. The integral condition is complicated partly by the large number of integral constraints. For instance, when the type space is the unit cube the boundary condition $\mu_1(t) = 0$ applies on two faces and the differential constraints apply on $m - 1$ additional faces, so there remain $m - 1$ faces where $[m - 1]n^{m-1}$ boundary values must be determined from integral constraints. For the standard test problem, the analog of the simple approximation of the Jacobian used for 2-dimension problems works equally well in three dimensions.

The main complication, however, is the fact that each integral constraint depends on $m - 1$ of the multipliers. For the standard test problem, at one of these values on the face where $t_1 = 0$, the integral constraint is

$$- \int_0^1 \left(f(t) + \sum_{i=2}^{m} \frac{\partial \mu_i}{\partial t_i}(t) \right) dt_1 = 0 ,$$

which involves more than one of the multipliers if $m > 2$. Consequently, it is apparently necessary to solve for all the multipliers simultaneously. In the next paragraphs, such a scheme is outlined for the general case, including nonlinear versions of the integrability constraint; the comments regarding Phase 2 in that case apply here as well.

The Appendix includes the APL computer program *nra3D* that implements this algorithm for the 3-dimensional version of the standard test problem. The approximation of the multiplier $\mu_1(t)$ obtained from this program is shown in Figure 5 for all values of t_1 and several values of the other two type parameters. Note that the two values of t_3 yield almost the same curve for $t_2 = 1$.

[14] Standard software is available for solving fairly general elliptic equations with Dirichlet boundary conditions of considerable complexity. One source of such software written in Fortran is NetLib at Oak Ridge National Laboratory: the Internet address is NetLib@ornl.gov.

Nonlinear Equations

No examples for which the integrability condition is nonlinear have been solved, but we hazard a guess about the modifications required. Phase 1 in such cases evidently requires one to solve the bundle-optimality condition (4), the welfare-optimality condition (5), and the integrability condition (7), jointly for the $2m$ values $\langle q_i(t), \mu_i(t)\rangle_{i=1,\ldots,m}$ at each grid point t. This involves $2m$ equations, of which all but (5) is typically nonlinear; consequently, one of the many algorithms for solving nonlinear equations must be used.

Based on experience from two-dimensional examples, the key modification required for convergence and stability is that the m equations derived from (5) and (7) should be formulated as the *sum* of the $2m$ sets of m equations derived from each of the $m+1$-point stencils obtained from rotations of a basic stencil. With this modification, Phase 1 is again a relaxation algorithm, albeit a nonlinear one, for all m multipliers simultaneously, and incidentally the m quantities, at each grid point.[15]

In Phase 2, the differential condition derived from the joint application of the transversality condition and the integrability condition is also nonlinear, so again a nonlinear equation must be solved. The integral condition requires no significant modification since it involves only the welfare-optimality condition, which is linear. However, it should be noted that, because a solution is sought for all multipliers simultaneously, at each point on *each* segment of the boundary one of the boundary values must be determined by either a differential and or an integral condition. As mentioned above, each integral condition requires summing over values of the discrete approximations of the derivatives of $m-1$ of the multipliers.

These considerations about the Newton-Relaxation Algorithm are considerably simplified if one uses the higher-dimensional version of the Pure Relaxation Algorithm, because the role of Newton's Method is replaced by using the welfare-optimality condition to specify the boundary conditions on the lower boundaries. Lacking computational experience with this algorithm in more than two dimensions, however, we are reluctant to venture a guess about its implementation.

Construction of the Price Schedules and the Tariff

The end product of the algorithm is an approximation of the vector multiplier $\mu(t)$ at each point t in a discrete grid $\hat{T}$ used to approximate the domain of types. Using

[15] Some discussion and results about methods and conditions for convergence and stability of nonlinear systems are included in Golub and Ortega (1992, §5.3).

this multiplier, one obtains the optimal assignment of the bundle $q(t)$ to type t from the ordinary equation (4). If this equation is nonlinear then one can use any standard method for solving nonlinear equations, such as Newton's method, or one of the standard software packages for nonlinear optimization. Where the bundle $q(t)$ is nonzero, say for $t \in \hat{T}_+$, the vector of marginal prices is required to be

$$\hat{p}(t) \equiv p(q(t)) = U_q(q(t), t),$$

at least for those components for which $q_i(t) > 0$, and the total tariff is

$$\hat{P}(t) \equiv P(q(t)) = U(q(t), t) - W(t),$$

where $W(t) = 0$ and $\hat{P}(t) = 0$ on the lower boundary of $\hat{T}_+$ where $q(t) = 0$.

The next step is to construct the actual vector $p(q)$ of marginal prices and then the total tariff $P(q)$. This can be done by Delaunay triangulation, which for the two-dimensional case is implemented in *Mathematica* (Wolfram, 1991; Boylan, 1991). From the list $\langle q_1(t), q_2(t), \hat{p}_1(t) \rangle$ for $t \in \hat{T}_+$, for instance, this technique produces a piecewise-linear approximation of $p_1(q)$ as a function of the vector $q = (q_1, q_2) \in Q$; and $p_2(q)$ can be approximated similarly. Figure 4 shows the Delaunay triangulation of the marginal price schedule $p_1(q)$ for the standard test problem. Alternatively, the isoquants of the marginal price schedules can be calculated by polynomial interpolation, as in Figure 3.

As mentioned earlier, if the bundle jumps upward from zero for the least types served, then it is necessary to include a fixed fee in the tariff to ensure that lower types find it unprofitable to purchase a positive bundle. This fee is the amount of the surplus otherwise gained by these types. Taking this fee into account, the penultimate step calculates the total tariff $P(q)$ as the path integral (approximated discretely) of the marginal prices from the origin to q. As a check that the integrability condition (7) is satisfied, it is useful to verify that the tariff is independent of the path used, or nearly so to the degree of the accuracy allowed by the numerical calculations.

This completes the construction of the solution to the relaxed problem in Mirrlees' formulation. In principle, however, a final step is required. One must still verify that the solution of the relaxed problem solves the complete problem. This involves two checks, one to verify that each type's bundle is globally optimal given the tariff, and another to verify that the tariff is a global solution of the seller's problem. Although sufficient conditions (and the ironing procedure for modifying the solution of the relaxed problem) ensuring these global properties are well-known for the one-dimensional case, comparable results for multidimensional cases are known only for formulations having the increasing-differences and super-modularity properties studied by Milgrom and Shannon (1994).

An Alternative Version

An alternative version of the algorithm is described by Wilson (1993, §13.6). Although it can be adapted to more general problems, we sketch it only for the standard test problem.

The idea is to reinterpret the second-order equation for μ_1 as a first-order equation for the two functions $a(t) \equiv -\frac{\partial}{\partial t_1}\mu_1(t)$ and $b(t) \equiv -\frac{\partial}{\partial t_2}\mu_1(t)$, linked by the integrability condition $\frac{\partial}{\partial t_2}a(t) = \frac{\partial}{\partial t_1}b(t)$. The boundary conditions imply that $b(t) = 0$ on the left and right boundaries where $t_1 \in \{0,1\}$, and $\int_0^1 a(t)\,dt_1 = 0$ for all values of t_2. In fact, one can show that if the latter condition is satisfied on the upper boundary where $t_2 = 1$ then it is satisfied everywhere, and therefore the boundary condition for b on the left boundary is extraneous. In a discrete version, therefore, one implements the algorithm by initially guessing values of a on the upper boundary that sum to zero, and specifying that the values of b on the right boundary are all zero. From these values one can then use the discrete approximations of the first-order equation and the integrability condition to calculate the values of a and b at all grid points from the 3-point stencil rotated $180°$, proceeding from upper-left to lower-right along successively lower diagonals. The aim, therefore, is to find values of a on the upper boundary that ensure that the integral condition is satisfied, expressed here in the form

$$\int_0^1 [f(t) - a(t)]\,dt_2 = 0$$

for each value of $t_1 > 0$, or the corresponding summation in discrete form. One can therefore use Newton's method (as above) to improve iteratively the estimates of a on the upper boundary.

One can do better than this, however, when the first-order equation is linear with constant coefficients. The solutions for a and b throughout the grid can be expressed in terms of the coefficients of the values of a from the upper boundary. Consequently, the integral conditions provide a set of linear equations, which fortunately is nonsingular. Solving these equations yields the required values of a on the upper boundary, and then the values of a and b throughout the grid can be obtained from the coefficients. Finally, one obtains the multiplier from the discrete version of the formula $\mu_1(t) = \int_{t_1}^1 a(t)\,dt_1$. The solution of the standard test problem obtained from this algorithm differs only slightly from the solutions produced by nonlinear optimization, Fourier approximation, and the relaxation algorithm.

The Appendix includes an APL program *aalg2D* that implements this algorithm for the standard test problem.

3. Summary and Conclusions

Nonlinear pricing is one instance of a wide variety of problems derived from the principal-agent paradigm and its extensions to mechanism design. The key ingredient of the standard formulation is the representation of an agent's private information as a point in a Euclidean space. The key step in characterizing a solution is reliance on the necessary conditions derived from the seller's relaxed problem. This approach is fruitful when the type space is one-dimensional because the Lagrange multiplier on the incentive-compatibility constraint can be obtained in closed form. When the type space is multidimensional, however, construction of the multipliers presents a classical problem involving first-order partial differential equations, complicated by awkward boundary conditions. The differential equations, moreover, involve both the simple linear form in the welfare-optimality condition (5) and possibly nonlinear forms derived from the integrability condition (7).

Computational procedures naturally divide into two phases. In Phase 1 the differential equations are solved with fixed (Dirichlet) boundary values. Straightforward approaches to solving these equations based on discrete approximations encounter inconsistencies. The explanation seems to be that the integrability condition is properly formulated in terms of backward derivatives and the welfare-optimality condition, in terms of forward derivatives. The Pure Relaxation Algorithm relies on this asymmetry to form the recursion (from a symmetric stencil) for successive relaxation on the interior. The more cautious Newton-Relaxation Algorithm enforces a symmetric setup by including all rotations of a basic stencil to form the recursion for successive relaxation. In some examples, inclusion of the rotated stencils is equivalent to solving the second-order equation derived from the welfare-optimality and integrability conditions.

In Phase 2 the boundary values are adjusted to improve the approximation of the boundary conditions imposed by the transversality condition. For both algorithms, the differential condition (derived from the integrability condition) provides an adequate (von Neumann) boundary condition. In the Pure Relaxation Algorithm, the boundary conditions on the lower boundaries can be derived from the welfare-optimality condition. This yields an especially simple scheme for calculations. In the Newton-Relaxation

Algorithm, however, these boundary conditions are specified by the integral constraints (derived by integrating the welfare-optimality condition), and Newton's method is used to obtain successive improvements. Fortunately, simple approximations of the Jacobian suffice for Newton's method in the limited class of examples that have been studied.

Limited experience with an alternative algorithm indicates that it may also be feasible to use an $m + 1$-point stencil if the functions are interpreted as the gradients of the multipliers, but this algorithm has been implemented only for the standard test problem.

The examples that have been solved indicate that these difficult computations are worthwhile. In particular, the qualitative properties of the tariff and its marginal prices bear little relation to those predicted from studies of the one-dimensional case. For example, in the standard test problem the multipliers are not monotone functions of the types, and the marginal prices of commodities are not monotone functions of the quantities purchased. Also, each customer purchases either both commodities or neither, which indicates that implicit 'bundling' is an essential ingredient, as Armstrong (1992, 1993) has emphasized. The implication is that the chief qualitative features of multi-product pricing and taxation differ substantially from the single-product case.

The complexity of the multidimensional nonlinear pricing problem addressed here suggests that an entirely different formulation might be useful in practice. An alternative approach is developed in Wilson (1993; §12, §14) by relying on a formulation in which it is supposed that the seller has no information about the distribution of types and about the dependence of customers' preferences on their types. Instead, the seller knows only the aggregate distribution of bundles purchased in response to linear tariffs. This formulation precludes an exact analysis of the participation constraint (3) because the demand data do not distinguish whether a price increase curtails a customer's demand or extinguishes participation; in particular, pure bundling is generally not optimal. On the other hand, this formulation allows calculation of a solution via a simple gradient algorithm, and quite complicated problems can be addressed routinely.

References

Armstrong, Mark (1992), "Optimal Nonlinear Pricing by a Multiproduct Monopolist," Chapter 4 of D. Phil. Thesis. Oxford, UK: Institute of Economics and Statistics of Oxford University.

Armstrong, Mark (1993), "Multiproduct Nonlinear Pricing," Economic Theory Discussion Paper #185, Cambridge University, UK.

Boylan, Philip (1991), "Discrete Mathematics," and the function *Triangular Surface Plot* in the program *Discrete Math 'Computational Geometry'*, Chapter 4 in *Guide to Standard Mathematica Packages*, Technical Report, Wolfram Research, Inc.

Derrick and Grossman (1987), *A First Course in Differential Equations with Applications*, Third Edition.

Golub, Gene H., and James M. Ortega (1992), *Scientific Computing and Differential Equations*. San Diego, CA: Academic Press, Inc.

Guesnerie, Roger, and Jean-Jacques Laffont (1984), "A Complete Solution to a Class of Principal-Agent Problems with an Application to the Control of a Self-Managed Firm," *Journal of Public Economics* 25: 329-369.

Milgrom, Paul R., and Christina Shannon (1994), "Monotone Comparative Statics," *Econometrica* 62: 157-180.

Milne, William E. (1970), *Numerical Solution of Differential Equations*, Second Edition. New York: Dover Publications.

Mirrlees, James A. (1971), "An Exploration in the Theory of Optimal Taxation," *Review of Economic Studies* 38: 175-208.

Mirrlees, James A. (1976), "Optimal Tax Theory: A Synthesis," *Journal of Public Economics* 6: 327-358.

Mirrlees, James A. (1986), "The Theory of Optimal Taxation," Chapter 24 in K.J. Arrow and M.D. Intriligator (eds.), *Handbook of Mathematical Economics*, Volume III, 1197-1249. New York: Elsevier Science Publishers.

Mussa, Michael, and Sherwin Rosen (1978), "Monopoly and Product Quality," *Journal of Economic Theory* 18: 301-317.

Roberts, Kevin W.S. (1979), "Welfare Considerations of Nonlinear Pricing," *Economic Journal* 89: 66-83.

Wilson, Robert (1993a), *Nonlinear Pricing*. New York: Oxford University Press.

Wilson, Robert (1993b), "Design of Efficient Trading Procedures," Chapter 5 in Daniel Friedman and John Rust (eds.), *The Double Auction Market: Institutions, Theories, and Evidence*, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Volume XIV, 125—152. Reading, MA: Addison-Wesley Publishing Company.

Wolfram, Stephen (1991), *Mathematica*, Second Edition. Redwood City, CA: Addison-Wesley Publishing Co.

Appendix

APL Programs

The APL programs *nra2L* and *nra3D* that implement the relaxation algorithm with Newton's method are shown on the following page. The program *nra2L* is designed for 2-dimensional problems with linear coefficients as in (8), whereas *nra3D* is designed for the 3-dimensional version of the standard test problem. Both use the inverse of the coarse approximation of the Jacobian (described in the text), expressed in terms of partial sums of the errors in the integral constraints.

On the following page are the APL programs *pra2D* and *pra2L* that implement the pure relaxation algorithm for the 2-dimensional version of the standard test problem, and the linear-coefficients model as in (8).

Shown next is the program *aalg2D* that implements the alternative algorithm for the standard test problem.

The parameter n in the programs plays the role of $n + 1$ in the text. All four programs assume that the domain of the type parameters is the unit square.

```
      ⎕cr 'nra2L'
mu1←C nra2L f;b;d;er;m;n;x;∆;∆∆;∆f;⍙
⍝∇ nra2L : Relaxation Algorithm : U_q=At-Bq, f arbitrary, t∊[0,1]^m. m=2.
⍝ Based on 5-point stencil, plus boundary conditions for mu1.    2=m=⍴⍴f.
⍝ Initial guess of mu1 is x; iterate from there.  2=⍴⍴C, 4=⍴C, C=A'(⌹B)A.
⍝ x=mu1[t1;t2]; t1 & t2∊{0,∆,2∆,...,1}; ∆[1 2]=1÷(m,n)-1.        ∆∆=●, ⍙=ε.
 ⍙←.001 ⋄ ∆∆←.2 ⋄ n←⍴f ⋄ x←n⍴0 ⋄ ∆←÷n-1 ⋄ m←n[1] ⋄ n←n[2] ⋄ C←,C ⍝Set up.
 b←(b,-C[2]÷2××/∆)÷d←2××/b←C[4 1]÷∆*2 ⋄ x[1;]←x[m;]←0        ⍝↓ PDE Coeffs.
 ∆f←(((C[4]÷∆[1])×1 ¯1↓(1 1↓f)-¯1 1↓f)-(C[2]÷∆[2])×¯1 1↓(1 1↓f)-1 ¯1↓f)÷d
L: d←∆[2]×+/0 ¯1↓-(¯1 0↓f)+((1 0↓x)-¯1 0↓x)÷∆[1] ⍝  Compute error at bdy.
 x[;1]←x[;1]-(∆∆×⌽+\⌽d),0 ⋄ x[;n]←x[;n-1]           ⍝ Apply Newton's Method.
M: mu1←(b[1]×0 ¯1↓(2 1↓x)+(¯2 1↓x))+b[2]×¯1 0↓(1 2↓x)+(1 ¯2↓x) ⍝   Solve
 mu1←mu1+(b[3]×(2 2↓x)+(¯2 ¯2↓x)-(2 ¯2↓x)+(¯2 2↓x))+∆f   ⍝ PDE via relax.
 mu1←x[;1],(0⍪mu1⍪0),x[;n] ⋄ →(⍙>⌈/er←(⌈/|d),(⌈/|,mu1-x))/0       ⍝ →Exit.
 x←mu1 ⋄ →(⍙<er[2])/M ⋄ →L                       ⍝ Repeat until error small.

      ⎕cr 'nra3D'
mu←nra3D N;d;dj;er;f;i;j;k;l1;l2;l3;m;n;o;p;q;r;x;∆;∆∆;⍙
⍝∇ nra3D : Relaxation Algorithm: standard test problem: U_q=t-q;f=1;m=3!
⍝ Initial guess of mu is x; iterate from there.  (3,N,N,N)≡⍴x, ⍙=ε, ∆∆=●.
 m←3 ⋄ x←(m,m⍴N)⍴0  ⍝ x=mu[i;t1;t2;t3]; t_i∊{0,∆,2∆,...,1}; ∆=1÷n, n=N-1.
 ⍙←.001 ⋄ ∆∆←.3÷m ⋄ ∆←÷n←(N←1↓⍴x)-1 ⋄ r←⍳m←⍴N ⋄ f←N⍴1          ⍝←↓ Set up.
 l1←1↓⍳n[1] ⋄ l2←1↓⍳n[2] ⋄ l3←1↓⍳n[3]
Phase2: ⍝* Improve boundary values via Newton Method & Differential Cond.
L: dj←(0,N)⍴0 ⋄ p←⌽m↑1 ⋄ j←1 ⍝ Calculate d=errors in Integral Conditions.
 p←¯1⌽p ⋄ d←((p↓x[j;;;])-(-p)↓x[j;;;])÷∆[j] ⋄ dj←dj⍪d,[j](p×N-2)↓d
 →(m≥j←j+1)/⎕lc-1 ⋄ d←(0,1↓N)⍴0 ⋄ q←⌽m↑¯1 ⋄ i←1
 q←¯1⌽q ⋄ d←d⍪-∆[1]×+/[1]q↓f++⌿dj[r~i;;;] ⋄→(m≥i←i+1)/⎕lc ⍝↓Newton Method.
 x[1;;1;]←x[1;;1;]-∆∆×⊖+⍀⊖d[2;;] ⋄ x[1;;;1]←x[1;;;1]-∆∆×⊖+⍀⊖d[3;;]
 x[2;1;;]←x[2;1;;]-∆∆×⊖+⍀⊖d[1;;] ⋄ x[2;;;1]←x[2;;;1]-∆∆×⌽+\⌽d[3;;]
 x[3;1;;]←x[3;1;;]-∆∆×⌽+\⌽d[1;;] ⋄ x[3;;1;]←x[3;;1;]-∆∆×⌽+\⌽d[2;;]
 x[2 3;N[1];;]←x[2 3;N[1]-1;;]    ⋄ x[1 3;;N[2];]←x[1 3;;N[2]-1;]
 x[1 2;;;N[3]]←x[1 2;;;N[3]-1] ⋄ d←(0,(m-1)⍴1)↓d ⍝↑ Different. Condition.
Phase1: ⍝* Solve PDE via Relaxation Algorithm using (2m+1)-point stencil.
 x[1;1;;]←x[1;N[1];;]←0 ⋄ x[2;;1;]←x[2;;N[2];]←0 ⋄ x[3;;;1]←x[3;;;N[3]]←0
M: o←¯1+q←⌽m↑1 ⋄ p←1+q ⋄ q←1-3×q ⋄ mu←0 ⋄ i←¯1    ⍝↓ Relaxation algorithm.
 mu←mu+(0,i⌽o)↓((0,i⌽p)↓x)+(0,i⌽q)↓x ⋄ →(m≥-i←i-1)/⎕lc ⍝ Approximate PDE.
 er←(⌈/|,d),⌈/|,x[;l1;l2;l3]-mu←mu÷2×m ⋄ x[;l1;l2;l3]←mu   ⍝ Update data.
 mu←x ⋄ →((⍙>⌈/er),⍙<er[2])/0,M ⋄ →Phase2    ⍝ Stop or repeat Relaxation.
```

```
      ⎕cr 'pra2D'
mu←pra2D n;A;a0;a1;b0;b1;ep;er;f;i;l;s;x;xn;y;yn;∆;∆∆
⍝∇ pra2D n : Pure relaxation algorithm for std test problem: U_q≡t-q,f≡1.
⍝ Uses 'backward' Integrability and 'forward' Welfare Optimality. 2≡m≡⍴n.
 f←n⍴1 ⋄ ep←.0001 ⋄ s←4+⌈/n ⋄ mu←(2,n)⍴0 ⋄ l←n-2 ⋄ x←y←l⍴0 ⋄ i←1
 a0←a1←l[1]⍴0 ⋄ b0←b1←l[2]⍴0 ⋄ ∆∆←((,∆∘.×∆),×/∆×∆)÷∆+.×∆←÷n-1  ⍝ PDE wts.
Phase1:                    ⍝ * Relaxation algorithm for interior grid points.
 A←(0 ¯1↓a0,x)⍪(0 1↓y,0)⍪(¯1 0↓b0⍪y)⍪(1 0↓x⍪0),[.5]1 1↓¯1 ¯1↓f
 xn←(∆∆×1 1 ¯1 1,n[1]-1)+.×A ⋄ yn←(∆∆[3 1 4 2 5]×¯1 1 1 1,n[2]-1)+.×A
 A←0 ⋄ →((30>i←i+1)∧ep<er←⌈/|,((x←xn),y←yn)-x,y)/Phase1 ⋄ xn←yn←⍬
Phase2: ⍝ * Improve boundary values via Differential, Welfare Optimality.
 a1←x[;l[2]] ⋄ b1←y[l[1];] ⋄ A←0 ⋄ i←l[1]        ⍝ ←Differential Condition.
 a0[i]←A←a0[i]-s×a0[i]-A+∆[1]×(y[i;1]+∆[2])+f[i+1;1] ⋄ →(1≤i←i-1)/⎕lc
 A←0 ⋄ i←l[2]   ⍝ ↑↓ s≡stepsize.          ⍝ ↑↓ Welfare Optimality Condition.
 b0[i]←A←b0[i]-s×b0[i]-A+∆[2]×(x[1;i]+∆[1])+f[1;i+1] ⋄ →(1≤i←i-1)/⎕lc
 i←1 ⋄ er←⌈/|,(mu←(0⍪(a0,x,a1)⍪0),[.5](0,(b0⍪y⍪b1),0))-mu
 ⍞←(4⍕er),(50<⎕cursor[2])⍴⎕tcnl ⋄ →(ep<er)/Phase1
```

```
      ⎕cr 'pra2L'
mu←C pra2L f;a;a0;a1;b;b0;b1;d;df;e;ep;er;g;h;i;j;l;n;s;u;v;x;y;F;V;W;∆
⍝∇ mu←C pra2L f : Pure relaxation algorithm for problem: U_q≡At-Bq, where
⍝∇ t∊[0,1]*2; C≡(⍉A)+.×(⌹B)+.×A & density f are arbitrary. 2 2≡⍴C, 2≡⍴⍴f.
⍝∇ Uses 'backward' Integrability and 'forward' Welfare Optimality.
 ep←.0001 ⋄ n←⍴f ⋄ ∆←(1 ¯1)×÷n-1 ⋄ s←4+⌈/n ⋄ l←n-2 ⋄ h←-g←1 1 ⋄ i←1
 mu←(2,n)⍴0 ⋄ x←y←l⍴0 ⋄ a0←a1←l[1]⍴0 ⋄ b0←b1←l[2]⍴0     ⍝ Initial guesses.
Wts: ⍝ * Compute weights for 4 points of stencil in relaxation algorithm.
 d←(⌽+/C×∆∘.×∆)+.×df←(h↓g↓¯1 0↓0⍪f),[.5]h↓g↓0 ¯1↓0,f ⋄ F←h↓g↓f
 W←(2 4⍴∆∘.×e←⍉⊖C×⍉2 2⍴∆)∘.×F÷d    ⍝ W≡wts applied to n11,.,n22,m11,m22,f.
 W←W,[2]1 3 4 2⍉(⊖(⍉C×⍉2 2⍴∆,-∆)+.×⊖df÷d,[.5]d)∘.×|(⌽∆),×/∆ ⋄ ∆←|∆
 F←(F,[.5]F)×W[;7;;] ⋄ W←0 ¯1 0 0↓W ⍝↓Weights for Differential Condition.
 g←⍉(,e)∘.×(¯1↓1↓f[;n[2]])÷e[1;]+.×df←(¯2↓  f[;n[2]]),[.5]¯1↓1↓f[;l[2]]
 h←⍉(,e)∘.×(¯1↓1↓f[n[1];])÷e[2;]+.×df←(¯1↓1↓f[l[1];]),[.5]¯2↓  f[n[1];]
Phase1:                    ⍝ * Relaxation algorithm for interior grid points.
 V←(¯1 0↓0⍪x)⍪(0 ¯1↓a0,x)⍪(¯1 0↓b0⍪y)⍪(0 ¯1↓0,y)⍪(1 0↓x⍪0),[.5](0 1↓y,0)
 u←F[1;;]++⌿W[1;;;]×V ⋄ v←F[2;;]++⌿W[2;;;]×V ⍝↑ Values at stencil points.
 V←⍬ ⋄ →((30>i←i+1)∧ep<er←⌈/|,((x←u),y←v)-x,y)/Phase1 ⋄ u←v←⍬
Phase2: ⍝ * Improve boundary values via Differential, Welfare Optimality.
 a←b←0 ⋄ i←l[1] ⋄ j←0                           ⍝ ↓ Differential Condition.
A: j←j+1 ⋄ a1[j]←b←g[j;]+.×b,x[j;l[2]],0,y[j;l[2]]
 a0[i]←a←a0[i]-s×a0[i]-a+∆[1]×(y[i;1]+∆[2])+f[i+1;1] ⋄ →(1≤i←i-1)/A
 a←b←0 ⋄ i←l[2] ⋄ j←0 ⍝ ↑ s≡stepsize!   ⍝ ↑ Welfare Optimality Condition.
B: j←j+1 ⋄ b1[j]←b←h[j;]+.×x[l[1];j],0,y[l[1];j],b
 b0[i]←a←b0[i]-s×b0[i]-a+∆[2]×(x[1;i]+∆[1])+f[1;i+1] ⋄ →(1≤i←i-1)/B
 i←1 ⋄ er←⌈/|,(mu←(0⍪(a0,x,a1)⍪0),[.5]0,(b0⍪y⍪b1),0)-mu
 ⍞←(4⍕er),(50<⎕cursor[2])⍴⎕tcnl ⋄ →(ep<er)/Phase1
```

```
      ⎕cr 'aalg2D'
mu1←aalg2D n;i;j;k;m;x;A;B;It;X;Δ;ΔΔ
⍝∇ Solve Mirrlees conditions: Example1: U_q=t-q, f=1; Example 2: f=*-÷/t.
⍝ Integrability condition is d\mu2/dt1=Δ×d\mu1/dt2, where Δ=1 for Ex.1
⍝   and Δ=(r2/r1)*2 for Ex.2, where r1=*-t1, d/dt1=d/dr1, etc.
⍝ Note: t2 decreases from 1 as i increases; t1 increases as j increases.
⍝ A & B are coefficients in ∇mu1 of x=mu1(.,1). Accuracy requires n≤20.
 m←n+1 ⋄ A←B←((n-1),m,n)ρ0 ⋄ A[;1 2;1]←¯1 ⋄ j←n
 A[j-1;1 2;j]←1 ⋄ →(1<j←j-1)/⎕lc ⋄ i←2 ⋄ It←n-1
Ex: ⍞←'Ex 1 or 2? [1 or 2] ' ⋄ →(~(X←⍎⎕inkey)∊1 2)/0 ⋄ Δ←ΔΔ←1
Iterate:        j←1⌈It ⋄ k←1
Calculate:      →k/Integrability
Consistency:    A[;i;j]←A[;i-1;j]+B[;i-1;j-1]-B[;i-1;j] ⋄          →Next
Integrability:  →(X=1)/I ⋄ Δ←((i-1)÷(n-j))*2 ⋄ ΔΔ←((i-2)÷(n-j))*2
I:              B[;i;j]←((ΔΔ×B[;i-1;j])-(A[;i;j]-A[;i;j+1]))÷Δ ⋄ →Next
Next: k←2|k+1 ⋄ →(k∧~(i=m)∨(j=n))/Calculate ⋄ i←i+1 ⋄ j←j+1
 →((i≤m)∧(j≤n))/Calculate ⋄             k←1 ⋄ i←2 ⋄ →(1≤It←It-1)/Iterate
 i←3-It ⋄ j←1 ⋄ A[;i;j]←A[;i-1;j]-B[;i-1;j] ⋄              →(i<m)/Iterate
                x←((n-1)ρm-1)⌹+≠¯1 1 0↓3 1 2⍉A ⋄ mu1←(⌽+\⌽x+.×A,0)÷n
 mu1[⍳m-1;1]←0 ⋄ mu1←⍉⊖mu1             ⍝ Now mu1's arguments are [t1;t2].
```

Figures

Figure 1 shows the changes in the integral values resulting from a perturbation of $\mu_1(t)$ on the lower boundary where $t_1 = 1/2$ and $t_2 = 0$. These changes indicate that the Jacobian can be approximated by a matrix with diagonal and super-diagonal elements that are 1 and -1 respectively, with other elements set to zero.

Figure 2 displays the multiplier $\mu_1(t)$ for several values of t_2 using the version (8) in which the diagonal and off-diagonal elements of A, B, and C are 1 and $1/2$ respectively. The data for the figure were computed using the relaxation algorithm combined with Newton's method, and the grid size was $\delta = 0.025$.

Figure 3 shows the marginal price $p_1(q)$ of product 1 derived from Figure 2 using polynomial interpolation.

Figure 4 shows the marginal price $p_1(q)$ of product 1 for the standard test problem, calculated via Delaunay triangulation.

Figure 5 shows the 3-dimensional standard test problem's multiplier $\mu_1(t)$ for several values of t_2 and t_3, computed using the relaxation algorithm combined with Newton's method, using the grid size $\delta = 0.025$. Note that when $t_2 = 1$ the two values of t_3 yield nearly the same curve.

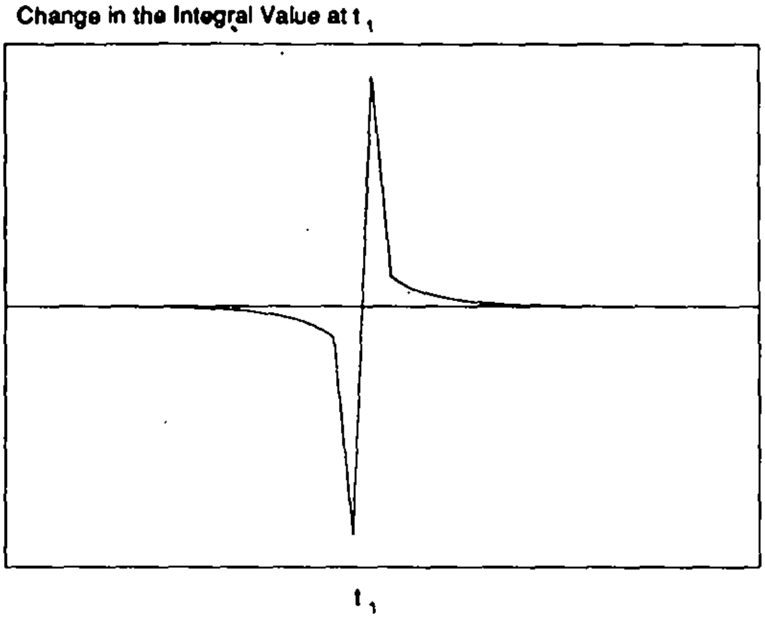

FIGURE 1 The changes in the integral values resulting from perturbing $\mu_1(1/2, 0)$ when $\mu_1(t) = 0$ initially. These changes, representing one column of the Jacobian matrix for the standard test problem, can be approximated by the function that is 1 and -1 at $t_1 = 1/2$ and $1/2 - \delta$, and zero elsewhere.

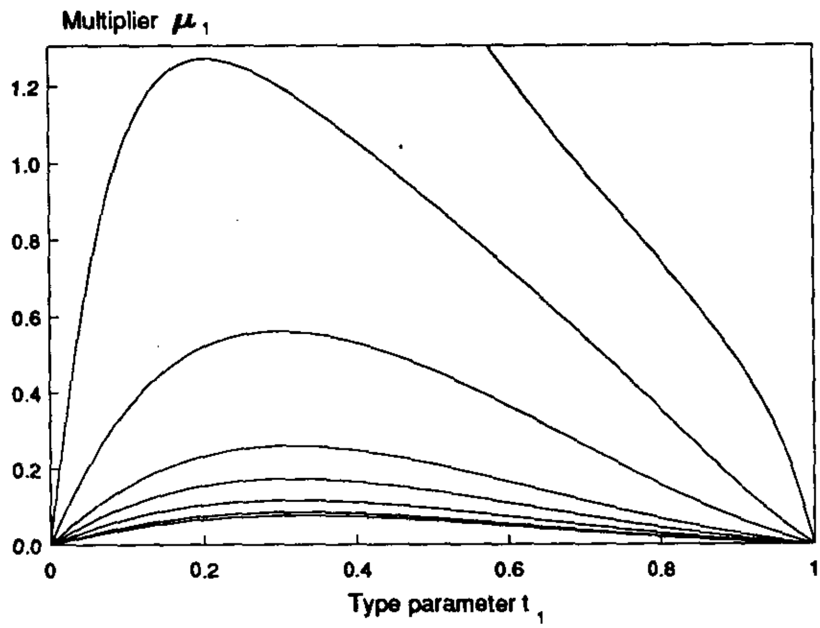

FIGURE 2 The multiplier $\mu_1(t)$ for the example based on (8) in which the diagonal elements of A, B, and C are 1 and their off-diagonal elements are $1/2$. From top to bottom, the curves correspond to $t_2 =$ 0, 0.2, 0.4, 0.6, 0.7, 0.8, 0.9, 1.0, although the bottom two are nearly indistinguishable. The mesh size $\delta = 0.025$ was used for the calculations.

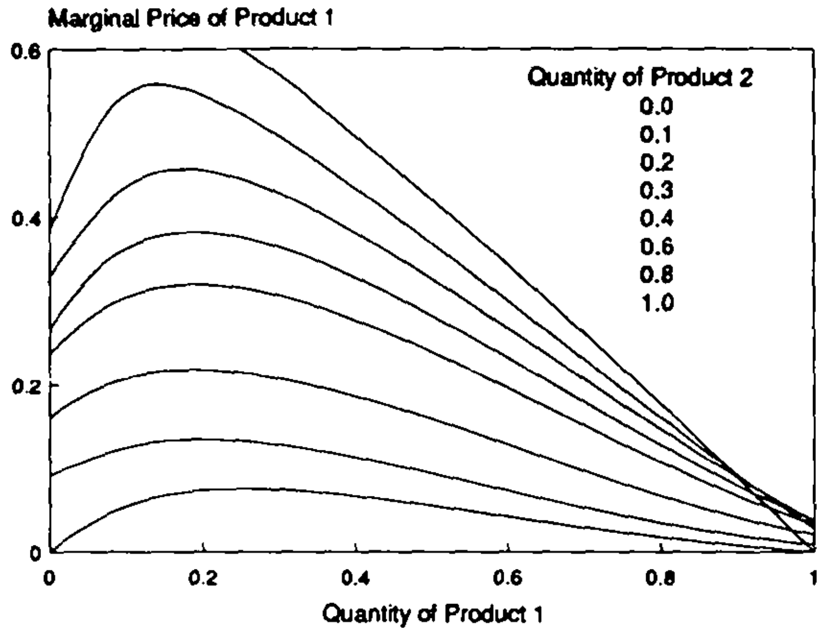

FIGURE 3 The schedule $p_1(q)$ of marginal prices for product 1 as a function of the vector $q = (q_1, q_2)$ of quantities of the two products. This approximation of the price schedule was derived from the data for Figure 2 using rational polynomial interpolation.

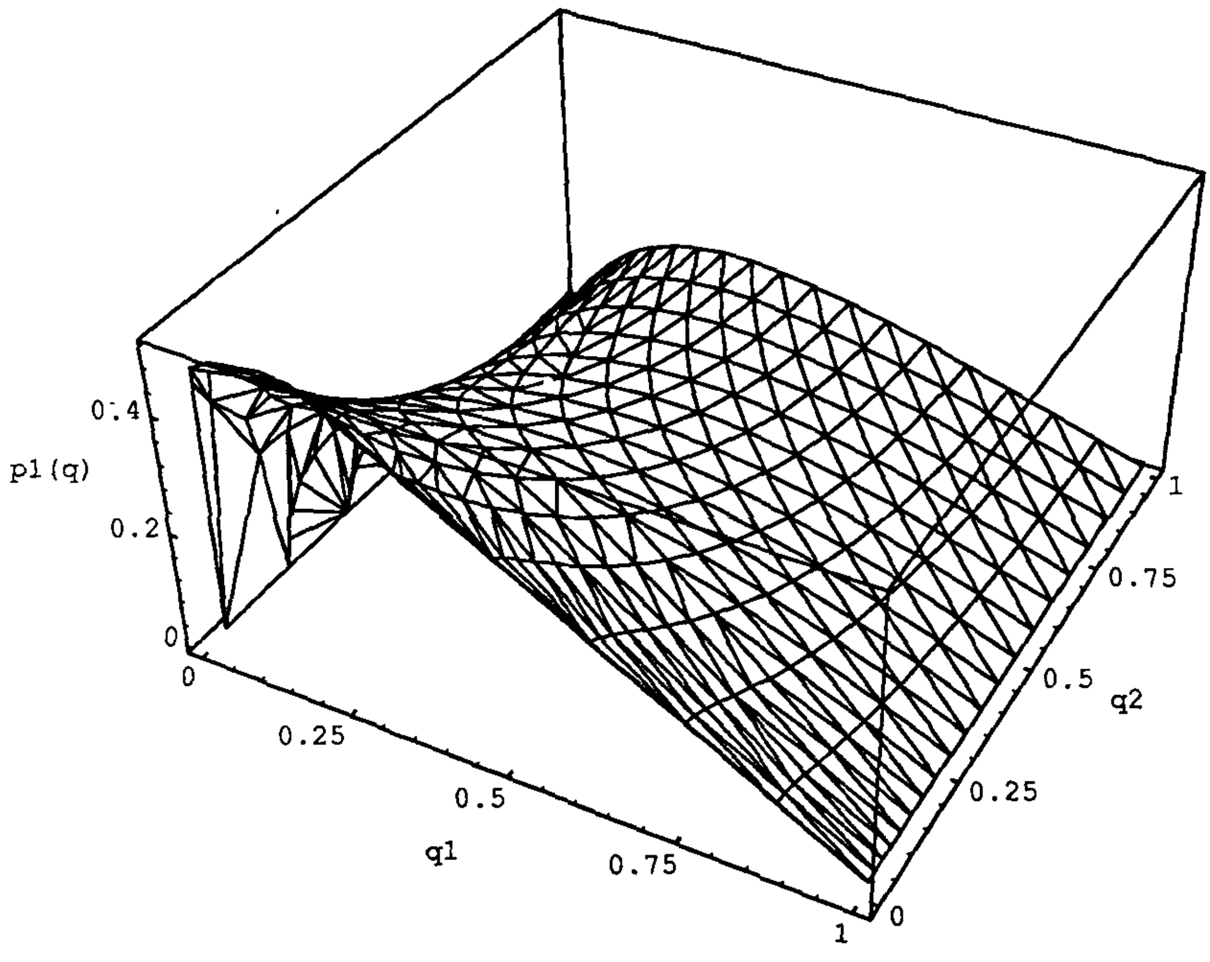

FIGURE 4 The schedule $p_1(q)$ of marginal prices for product 1 as a function of the vector $q = (q_1, q_2)$ of quantities of the two products. This piecewise-linear approximation of the price schedule for the standard test problem was constructed using Delaunay triangulation.

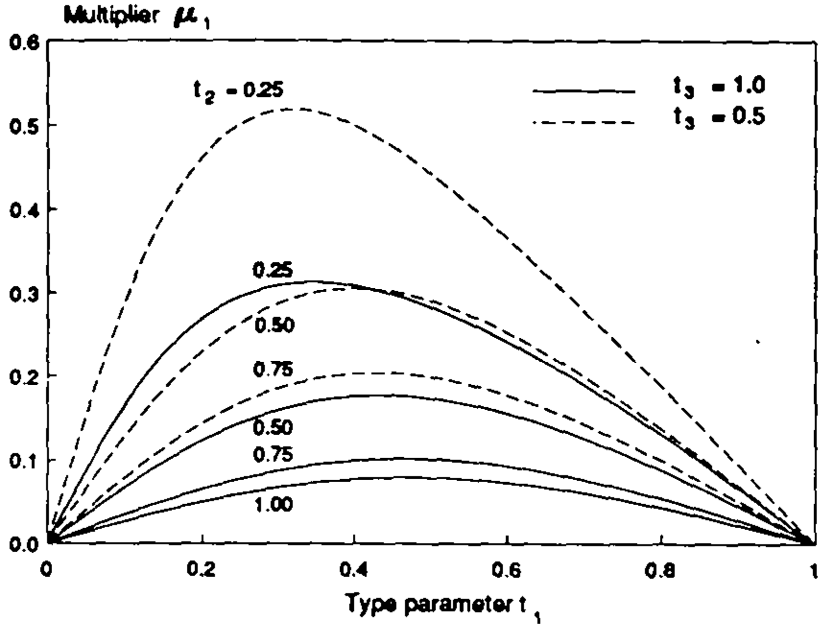

FIGURE 5 The multiplier $\mu_1(t)$ for the 3-dimensional version of the standard test problem, shown for the types $t_2 = 0.25, 0.50, 0.75, 1.00$, and $t_3 = 0.5, 1.00$. The bottom two curves are indistinguishable.