Mixed Complementarity Programming with GAMS

Thomas F. Rutherford
Lecture Notes for Econ 6433

Spring Semester, 2002

1 Introduction

GAMS (General Algebraic Modelling System) is a computer language which was originally developed to assist economists at the World Bank in the quantitative analysis of economic policy questions (Meeraus (1983), Brooke, Kendrick and Meeraus (1987)). At the time of its conception and initial development, linear programming (LP) was the standard framework in which market equilibrium models were constructed. Subsequently, solution algorithms and codes for general nonlinear programming representations became more common and alternative economic paradigms were developed. [1]

During this time, GAMS was extended from an LP matrix-generator to accommodate general nonlinear functions. GAMS was awarded the Computer Science Technical Section prize by the Operations Research Society of America in 1987. The release of a PC-based version of GAMS in 1988 lead to its wide-spread recognition as an effective tool for the development of general purpose large-scale mathematical programming models.

This paper introduces new features of the GAMS modelling language which permit formulating economic equilibrium models as systems of nonlinear equations, complementarity problems or variational inequalities. These extensions accommodate market and game-theoretic equilibrium models which are not easily studied in an optimization context. These formulations are here referred to as "mixed complementarity problems" (MCPs), reflecting the fact that they may incorporate mixtures of equations and inequalities. [2]

The MCP is a natural format for expressing a variety of economic models for both markets and games. Computational evidence suggests that algorithms for solving MCPs are relatively reliable and efficient, particulary for models which are not natural optimization problems.

The development of the GAMS/MCP modelling format was motivated by recent theoretical and practical developments in algorithms for nonlinear complementarity problems and variational inequalities. Some of these algorithms are extensions of classical methods for nonlinear equations. Others exploit the variational structure which is characteristic of many economic equilibrium models (see Dafermos (1983)). The most recent techniques are based on ideas from interior-point algorithms for linear programming (Kojima, Megiddo, Noma and Yoshise (1991)). A survey of developments in the theory and application of these methods is provided by Harker and Pang (1990).

At this time, two large-scale solvers are available through GAMS/MCP. MILES (a Mixed Inequality and nonLinear Equation Solver) employs a modified Newton algorithm which was originally developed for applied general equilibrium modelling (see Rutherford (1987, 1993), and Anstreicher, Lee and Rutherford (1992)). PATH is a recently developed solver based on a promising new path-following procedure (Dirkse and Ferris (1993), Ralph (1994)). Both codes incorporate implicit

[1] A chronology of these developments might include Samuelson (1952), Gale (1960), Takayama and Judge (1971), Goreux and Manne (1973) and Adelman and Robinson (1978).

[2] van der Laan and Tallman (1985) refer to these as "generalized complementarity problems".

bounds, mixtures of constrained and unconstrained variables, sparse matrix algebra and dynamic memory allocation. Both solvers employ the basis factorization package from MINOS 5.4, LUSOL (Gill et al.(1991)).

MILES executes a generalized Newton algorithm with a backtracking line search. This method is based on an algorithm investigated by Mathiesen (1985) who proposed a modeling format and sequential method for solving economic equilibrium models. Mathiesen's method is closely related to algorithms proposed by Robinson (1972), Hogan (1975), Eaves (1978) and Josephy (1979). The algorithm executed by PATH is a "global Newton" method in which the backtracking line search is replaced by a "path search". The algorithms share the same quadratic rate of convergence near a solution, but they may follow different trajectories away from the equilibrium. Because convergence cannot always be guaranteed with either algorithm, it is helpful to have both algorithms available when working with large or difficult problems.

There are two methods for expressing mixed complementarity problems in GAMS. The first is based on "standard" GAMS algebra with extended syntax to signify complementarity links between equations and variables. The second method is a GAMS interface for general equilibrium modeling with MPSGE (a Mathematical Programming System for General Equilibrium). MPSGE is a function and Jacobian evaluation program for a broad class of economic models. Nonlinear equations in an MPSGE model are automatically generated from a tabular description of cost and expenditure functions. The MPSGE interface is well suited for a specific class of functions, whereas GAMS/MCP can be applied to any complementarity problem which can be written with standard GAMS algebra. Any MPSGE model can be written in GAMS/MCP but not all GAMS/MCP models can be cast using MPSGE functions.[3]

The remainder of this paper is organized as follows. Section 2 defines the general class of mathematical programs which can be cast in an MCP format, Section 3 presents MCP formulations for three models which arise in market equilibrium analysis and game theory and computational benchmarks for MILES and PATH. Section 4 describes how local sensitivity analysis can be conducted within GAMS by using a nonlinear optimizer in conjunction with GAMS/MCP. Section 5 summarizes of GAMS/MCP syntax, and section 6 provides a brief conclusion. An appendix (available upon request) contains program listings for models described in the paper.

2 Manifestations of the Mixed Complementarity Problem

The "mixed-complementarity problem" (MCP) is defined here as:

$$\text{Given:} \qquad F : R^N \to R^N, \quad \ell, u \in R^N \qquad \text{(MCP)}$$

$$\text{Find:} \qquad z, w, v \in R^N$$

$$\text{s.t.} \qquad F(z) - w + v = 0$$

$$\ell \le z \le u, \quad w \ge 0, \quad v \ge 0$$

$$w^T(z - \ell) = 0, \quad v^T(u - z) = 0$$

in which $-\infty \le \ell \le u \le +\infty$. F must be continuously differentiable in order to express a model using GAMS/MCP algebra. Sufficient conditions for convergence of a Newton-type algorithm place additional restrictions on F. Once such condition would be that F is a P-function (see Harker and Pang (1990).)[4]

[3] The present paper does not present details of the MPSGE syntax. The interested reader is referred to Rutherford (1992b). MPSGE accommodates "auxiliary constraints" written in GAMS/MCP algebra for economic features which fall outside the standard Arrow-Debreu framework.

[4] In practice, it is quite difficult to determine whether the properties of a particular mapping guarantee convergence. For applied work, it is normal practice to first confirm convergence for a small-scale prototype and then

This MCP format encompasses a number of special cases, including the following:

(i) *a linear system of equations*:

$$\begin{array}{ll} \text{Given:} & A \in R^{n\times n}, \quad b \in R^n \quad \text{(LSYS)} \\ \text{Find:} & x \in R^n \\ \text{s.t.} & Ax = b \end{array}$$

which is represented as an MCP by letting $\ell = -\infty, \quad u = +\infty, \quad z = x, \quad$ and $F(z) = Az - b$;

(ii) *a nonlinear system of equations*:

$$\begin{array}{ll} \text{Given:} & f : R^n \to R^n, \quad b \in R^n \quad \text{(NLSYS)} \\ \text{Find:} & x \in R^n \\ \text{s.t.} & f(x) = 0 \end{array}$$

which is represented as an MCP by letting $\ell = -\infty, \quad u = +\infty, \quad z = x, \quad$ and $F(z) = f(z)$;

(iii) *a linear complementarity problem*:

$$\begin{array}{lll} \text{Given:} & M \in R^{n\times n}, \quad q \in R^n & \text{(LCP)} \\ \text{Find:} & z \in R^n & \\ \text{s.t.} & q + Mz \geq 0, \quad z \geq 0, \quad z^T(q + Mz) = 0 & \end{array}$$

which is represented as an MCP by letting $\ell = 0, \quad u = +\infty, \quad$ and $F(z) = q + Mz$;

(iv) *a nonlinear complementarity problem*:

$$\begin{array}{lll} \text{Given:} & f : R^n \to R^n & \text{(NCP)} \\ \text{Find:} & z \in R^n & \\ \text{s.t.} & f(z) \geq 0, \quad z \geq 0, \quad z^T f(z) = 0 & \end{array}$$

which is represented as an MCP by setting $\ell = 0, \quad u = +\infty, \quad$ and $F(z) = f(z)$;

(v) *a nonlinear program:*

$$\begin{array}{ll} \text{Given:} & f : R^n \to R, \quad g : R^n \to R^m, \quad \hat{\ell}, \hat{u} \in R^n \quad \text{(NLP)} \\ \text{Find:} & x \in R^n \text{ to} \\ & \max f(x) \\ \text{s.t.} & g(x) = 0 \\ & \hat{\ell} \leq x \leq \hat{u} \end{array}$$

which (when $f()$ is concave and $g()$ is convex) may be represented as an MCP by setting $N = n+m$, and partitioning[5]:

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \ell = \begin{pmatrix} \hat{\ell} \\ -\infty \end{pmatrix}, \quad u = \begin{pmatrix} \hat{u} \\ +\infty \end{pmatrix}, \quad F(z) \;= \begin{cases} \nabla f(x) - \nabla g(x)^T y \\ \qquad g(x) \end{cases}$$

proceed to develop a large scale model.

[5] y corresponds to the Lagrange multipliers on the constraints in the nonlinear program.

;

(v) *a finite-dimensional system of variational inequalities*:

$$\text{Given:} \qquad f : R^n \rightarrow^n R, \quad g : R^n \rightarrow R^m \qquad \text{(VIP)}$$

$$\text{Find:} \qquad x^* \in X \equiv \{\xi \in R^n | g(\xi) \geq 0\}$$

$$\max\, f(x^*)^T(x - x^*) \geq 0 \quad \forall x \in X$$

which (when $f()$ is convex and $g()$ is concave) is represented as an MCP by setting $N = n + m$, and partitioning[6]

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \ell = \begin{pmatrix} -\infty \\ 0 \end{pmatrix}, \quad u = (+\infty), \quad F(z) \; = \begin{cases} f(x) - \nabla g(x)^T y \\ g(x) \end{cases}$$

.

When a model can be directly expressed as a linear or nonlinear program, it is typically more efficient and reliable to apply a linear or nonlinear programming algorithm, several of which are available through GAMS. MCP is particularly useful for mathematical programs which cannot be processed as optimization problems. Local sensitivity techniques for MCP models may be employed for any of these mathematical programs.

3 Spatial Equilibrium Models

As an introduction to the MCP format, consider the Hitchcock-Koopmans transportation problem as described by Dantzig (1963). The data include a set of suppliers I and a set of markets J, with supplies a_i, demands b_j, transport costs c_ij from supplier i to market j. Figure 1 presents GAMS code defining the sets and data parameters used in the various models which are to be subsequently presented. Cast as a planning problem, this linear program seeks a transport schedule which minimizes the cost of supplying all markets. That is:

$$\min \; \sum_{ij} c_{ij} x_{ij}$$

$$\text{s.t.} \qquad \sum_j x_{ij} \leq a_i$$

$$\sum_i xij \geq b_j$$

$$x \geq 0$$

The GAMS code for the linear programming formulation is in Figure 2.[7]

It is well known that this problem can be interpreted as a market equilibrium problem in which the dual multiplier for an equation from the first set of constraints, w_i, represents the price in supply market i, and the dual multiplier for an equation in the second set of constraints, p_j, represents the price in demand market j.

[6] Observe that x^* in (VIP) solves:

$$\min_{x \in X} f(x^*)^T x$$

Provided that the relevant convexity and regularity conditions are satisfied, the MCP system constitutes necessary and sufficient conditions for minimization, and therefore an MCP solution is a solution to the original system of variational inequalities.

[7] This and subsequent Figures contain GAMS program fragments which illustrate ideas. Full program listings are provided in Appendix B. The MCP transportation model is in the GAMS model library. To copy it to your current directory execute "`gamslib transmcp`". Readers unfamiliar with the rudiments of the GAMS language may refer to Chapter 2 of the GAMS Users Guide (1987).

```
SETS
     I   canning plants   / SEATTLE, SAN-DIEGO /
     J   markets          / NEW-YORK, CHICAGO, TOPEKA / ;

PARAMETERS
      A(I)  capacity of plant i in cases (when prices are unity)
       /    SEATTLE     325
            SAN-DIEGO   575  /,
      B(J)  demand at market j in cases (when prices equal unity)
       /    NEW-YORK    325
            CHICAGO     300
            TOPEKA      275  /,

      ETA(I)  Price elasticity of supply
       /    SEATTLE     1.0
            SAN-DIEGO   1.0  /,

      SIGMA(J)  Price elasticity of demand
       /    NEW-YORK    1.5
            CHICAGO     1.2
            TOPEKA      2.0  /;

TABLE DISTANCE(I,J)  distance in thousands of miles
                  NEW-YORK       CHICAGO      TOPEKA
    SEATTLE          2.5           1.7          1.8
    SAN-DIEGO        2.5           1.8          1.4  ;

SCALAR F  freight in dollars per case per thousand miles  /90/ ;

PARAMETER C(I,J)  transport cost in thousands of dollars per case ;

C(I,J) = F * DIST(I,J) / 1000 ;

PARAMETER PBAR(J) Reference price at demand node J (suppy price = 1)
      /       NEW-YORK             1.225
              CHICAGO              1.153
              TOPEKA               1.126 /;
```

Figure 1: GAMS Set and Data Statements for the Transportation Model

```
  PARAMETER     ALPHA(I)  Supply function share coefficient (MCP),
                BETA(J)   Demand function share coefficient (MCP),
                GAMMA(I)  Supply function share coefficient (NLP),
                KAPPA(J)  Demand function share coefficient (NLP),
                EPSD(J)   Demand function elasticity coefficient,
                EPSY(I)   Supply function elasticity coefficient;

ALPHA(I) = A(I);
BETA(J)  = B(J) * PBAR(J)**SIGMA(J);
GAMMA(I) = (ETA(I) / (1 + ETA(I))) * (1/ALPHA(I))**(1/ETA(I));
KAPPA(J) = (SIGMA(J) / (SIGMA(J)-1)) * BETA(J)**(1/SIGMA(J));
EPSD(J) = (SIGMA(J) - 1) / SIGMA(J);
EPSY(I) = (ETA(I) + 1) / ETA(I);
```

Figure 2: GAMS Parameter Declarations and Assignments for the Transportation Model

```
VARIABLES
                X(I,J)          SHIPMENT QUANTITY FROM I TO J
                COST            MINIMAND - TOTAL COST OF SHIPMENT

POSITIVE VARIABLE X;

EQUATIONS
                SUPPLY(I)       SUPPLY LIMIT
                DEMAND(J)       DEMAND CONSTRAINT (FIXED)
                OBJDEF          DEFINES COST;

SUPPLY(I)..     A(I) =G= SUM(J, X(I,J));

DEMAND(J)..     SUM(I, X(I,J))  =G=  B(J);

OBJDEF..        COST =E= SUM((I,J), C(I,J) * X(I,J));

MODEL MINCOST / SUPPLY, DEMAND, OBJDEF/;

SOLVE MINCOST USING LP MINIMIZING COST;
```

Figure 3: The Transportation Model Formulated as a Linear Program

```
POSITIVE VARIABLES
                W(I)           SHADOW PRICE AT SUPPLY NODE I,
                P(J)           SHADOW PRICE AT DEMAND NODE J,
                X(I,J)         SHIPMENT QUANTITIES IN CASES;

EQUATIONS
                SUPPLY(I)      SUPPLY LIMIT AT PLANT I,
                DEMAND(J)      FIXED DEMAND AT MARKET J,
                PROFIT(I,J)    ZERO PROFIT CONDITIONS;

PROFIT(I,J)..   W(I) + C(I,J)   =G= P(J);

SUPPLY(I)..     A(I) =G= SUM(J, X(I,J));

DEMAND(J)..     SUM(I, X(I,J))  =G=  B(J);


MODEL TRNSP / PROFIT.X, SUPPLY.W, DEMAND.P/ ;

SOLVE TRNSP USING MCP;
```

Figure 4: The Transportation LP Formulated as an LCP

Cast as a market equilibrium problem, a system of IJ inequalities and associated complementary slackness conditions replace the minimization operator. "Zero profit conditions" for transportation activities characterize a competitive, constant-returns-to-scale (CRTS) "transportation" industry in which free entry drives excess profits to zero for active trade links and no unprofitable activities are operated. The equations which characterize this equilibrium are:

$$\sum_j x_{ij} \le a_i, \quad w_i \ge 0, \quad w_i \left(a_i - \sum_j x_{ij} \right) = 0 \quad \forall i$$

$$\sum_i x_{ij} \ge b_j, \quad p_j \ge 0, \quad p_j \left(b_j - \sum_i x_{ij} \right) = 0 \quad \forall i$$

$$w_i + c_{ij} \ge p_{ij}, \quad x_{ij} \ge 0, \quad x_{ij} \left(w_i + c_{ij} - p_j \right) = 0 \quad \forall i, j$$

This system of inequalities is a *linear* complementarity problem which may be specified in GAMS/MCP syntax as shown in Figure 3.

There would be no particular reason to formulate a linear program as an MCP. Suppose, however, that demands and supplies were price responsive and all markets perfectly competitive. These assumptions alone would not rule out an optimization approach, but they would require use of a *nonlinear* programming algorithm. For concreteness, let demand and supply functions be isoelastic. The revised equilibrium conditions are:

$$\sum_j x_{ij} \le \alpha_i w_i^{\eta_i}, \quad w_i \ge 0, \quad w_i \left(\alpha_i w_i^{\eta_i} - \sum_j x_{ij} \right) = 0 \quad \forall i$$

$$\sum_i x_{ij} \ge \beta_j p_j^{-\sigma_j}, \quad p_j \ge 0, \quad p_j \left(\beta_j p_j^{-\sigma_j} - \sum_i x_{ij} \right) = 0 \quad \forall i$$

$$w_i + c_{ij} \ge p_{ij}, \quad x_{ij} \ge 0, \quad x_{ij} \left(w_i + c_{ij} - p_j \right) = 0 \quad \forall i, j$$

This is a nonlinear complementarity problem.

When the matrix of cross-price elasticities for a partial equilibrium model is symmetric, there is an associated optimization problem which can be used to compute the equilibrium prices and quantities. (The model is said to be *integrable* - see Takayama and Judge (1971).) In the present example, cross-price elasticities are zero, and the associated optimization problem is:

$$\max \ \sum_i \gamma_i y_i^{\epsilon_i^y} + \sum_j \kappa_j d_j^{\epsilon_j^d} - \sum_{ij} c_{ij} x_i j$$

$$\text{s.t.} \qquad \sum_j x_{ij} \leq y_i$$

$$\sum_i x_{ij} \geq d_j$$

$$x_{ij} \geq 0, \quad d_j \geq 0, \quad y_i \geq 0$$

where:

$$\gamma_i = \frac{\eta_i}{1+\eta_i}\left(\frac{1}{\alpha_i}\right)^{1/\eta_i}, \quad \kappa_j = \frac{\sigma_j}{1+\sigma_j}\beta_j^{1/\sigma_j}, \quad \epsilon_i^y = \frac{1+\eta_i}{\eta_i}, \quad \text{and } \epsilon_j^d = \frac{\sigma_j - 1}{\sigma_j}$$

In economic terms, the market allocation “maximizes the sum of producer and consumer surplus”.

Economic equilibrium models are typically used to assess the consequences of market distortions, typically in the form of taxes, tariffs or other types of government policies. In public finance, equilibrium models are often used to measure the “excess burden” of a given tax structure. When *ad-valorem* taxes are present, the NLP formulation of a market equilibrium problem is not straight-forward. No single optimization problem characterizes the equilibrium because the resulting allocation is inefficient. Such an equilibrium could be computed by solving a *sequence* of nonlinear programs, but in these cases the MCP formulation is certainly more transparent.

As illustration of a model which is difficult to treat in an optimization framework but easy to pose as an MCP, consider applying *ad-valorem* taxes on trade flows in the spatial equilibrium model from above. This causes the supply price to be a non-unitary multiple of the marginal cost of supply, destroying integrability (notice that *specific* taxes do not cause this problem, as they can simply be added to the transport cost coefficients). The GAMS/MCP model is shown in Figure 4.

There are several directions in which the spatial price equilibrium model can be extended. Harker (1986) describes several spatial equilibrium models with imperfect competition. The variational and complementarity versions of these models are implemented in GAMS library models HARKER and HARKERMCP, respectively.

3.1 An MCP Formulation for N-Player Non-cooperative Games

The Lemke-Howson (1964) algorithm finds equilibria for two-player non-cooperative games with finite action spaces in pure or mixed strategies. An N-player extension of bimatrix games is conveniently expressed with the following notation:

- j denotes players $j \in \{1 \dots N\}$
- i denotes pure strategies $i \in \{1 \dots M\}$ (the same number of strategies are assumed available to all players to simplify notation)
- s denotes a “pure strategy profile” in which one pure strategy is assigned to each player, $s \in \{1 \dots M^N\}$
- $i(s, j)$ denotes the pure strategy assigned to player j in strategy profile s.
- $a_j s$ denotes the payoff to player j which arises from strategy profile s.

```
PARAMETER       T(I,J)          AD-VALOREM TAX RATE;

POSITIVE VARIABLES
                W(I)            SHADOW PRICE AT SUPPLY NODE I,
                P(J)            SHADOW PRICE AT DEMAND NODE J,
                X(I,J)          SHIPMENT QUANTITIES IN CASES;

EQUATIONS
                SUPPLY(I)       SUPPLY LIMIT
                DEMAND(J)       DEMAND CONSTRAINT (FIXED)
                PROFIT(I,J)     ZERO PROFIT CONDITIONS;


SUPPLY(I)..     ALPHA(I) * W(I)**ETA(I) =G= SUM(J, X(I,J));

DEMAND(J)..     SUM(I, X(I,J)) =G= BETA(J)  * P(J)**(-SIGMA(J));

PROFIT(I,J)..   (1 + T(I,J)) * (W(I) + C(I,J))  =G= P(J);

MODEL TRNSP / PROFIT.X, SUPPLY.W, DEMAND.P/ ;

* Avoid function evaluation errors by installing positive bounds:

P.LO(J) = 0.001;  W.LO(I) = 0.001;

* 10% tax on all trade:

T(I,J) = 0.10;

SOLVE TRNSP USING MCP;
```

Figure 5: Spatial Price Equilibrium with Ad-Valorem Taxes

Each player j chooses a vector of probabilities, p_{ij}, representing the probability of playing pure strategy i in the equilibrium. By definition:

$$p_{ij} \geq 0 \quad \forall i, j, \qquad \text{and} \ \sum_i p_{ij} = 1 \quad \forall j$$

.

Letq_{js} denote the probability that player j is confronted by pure strategy profile s in the equilibrium. Formally:

$$q_{js} = \Pi_{j' \neq j} p_{i(j',s),j'}$$

We assume that player j chooses a strategy profile to maximize the expected payoff, taking actions of the other players as given:

$$\max_{p_{ij}} \pi_j = \textstyle\sum_i p_{ij} \left(\sum_{s|i(s,j)=i} a_{js} q_{js} \right)$$

$$\text{s.t.} \qquad \textstyle\sum_i p_{ij} = 1, \quad p_{ij} \geq 0$$

A workable MCP formulation for the equilibrium problem needs to incorporate individual simplex constraints in a way which does not render the mapping singular at the equilibrium. In the formulation presented here, an additional variable y_j is associated with the simplex constraint for player j. It is easy to show that in any solution to this MCP, y_j must have a value of unity. The MCP formulation is:

$$\pi_j = \textstyle\sum_s a_{js} \left[\Pi_{j'} \left(\frac{p_{i(s,j'),j'}}{y_{j'}} \right) \right] \quad \forall j$$

$$\pi_j \geq \textstyle\sum_{s|i(s,j)=i} a_{js} \left(\Pi_{j' \neq j} \frac{p_{i(s,j'),j'}}{y_{j'}} \right), \quad p_{ij} \geq 0,$$

$$p_{ij} \left\{ \pi_j - \textstyle\sum_{s|i(s,j)=i} a_{js} \left(\Pi_{j' \neq j} \frac{p_{i(s,j'),j'}}{y_{j'}} \right) \right\} = 0, \quad \forall i, j$$

$$\textstyle\sum_i p_{ij} \geq 1, \quad y_j \geq 0, y_j \left(1 - \sum_i p_{ij}\right) = 1 \quad \forall j$$

The GAMS implementation of this model is displayed in Figures 7 and 8. Figure 7 sets up the data structures using dynamic sets to produce the mapping $i(s,j)$.

This is the "trickiest" part of the programming because it relies on the use of dynamic ordered sets. Once the data structure `LINK(S,I,J)` has been installed, it is straightforward to formulate the equilibrium conditions shown in Figure 8.

4 Local Sensitivity Analysis

Suppose that GAMS/MCP has been used to solve a nonlinear system of the form:

$$F(z; t) = 0$$

where z is an N-vector of decision variables, t is an M-vector of parameters, and F is a function mapping R^{N+M} to R^N.

Given a solution z^*, it is often helpful to know the local dependence of z^* on t.

Let J_z denote the Jacobian of F with respect to z (i.e. $\nabla_z F$), and let J_t denote the Jacobian of F with respect to t (i.e. $\nabla_t F$). It is easy to show that:

$$dz = -J_z^{-1} J_t dt \equiv S dt$$

where J_z^{-1} is the inverse of J_z. This equation defines the "local sensitivity matrix", $S = -J_z^{-1} J_t$. If the system of equations is specified so that J_z has full rank at the solution, then the gradients fall out as the solution of a linear system of equations. Setting up the linear system with the Jacobian

```
SET     J               PLAYERS /J1*J3/,
        I               ACTIONS /I1*I2/,
        SS              STRATEGY PROFILES /S1*S1000/,
        S(SS)           ACTIVE STRATEGY PROFILES,
        LINK(SS,I,J)    CONNECTION BETWEEN ACTIONS AND PROFILES;

ABORT$(CARD(SS) LT CARD(I1)**3) " SET SS IS NOT LARGE ENOUGH.";

PARAMETER
        PAYOFF(SS,J)    PAYOFF DATA IN STRATEGY-PROFILE FORMAT;

*       READ THE DATA:

TABLE LOSS(J,I,I,I)  PAYOFF DATA IN DENSE FORM

        I1.I1   I1.I2   I2.I1   I2.I2
J1.I1   1       2       8       5
J1.I2   8       8       2       2
J2.I1   4       2       2       1
J2.I2   2       6       1       1
J3.I1   4       1       4       2
J3.I2   8       8       2       1;


*       LET I1, I2 AND I3 ALL STAND FOR SET I:

ALIAS (I,I1),(I,I2),(I,I3);

*       ASSIGN PLAYER ACTIONS TO STRATEGY PROFILES USING
*       DYNAMIC SET S().

*       INITIALIZE SET S WITH A SINGLE ENTRY, THE FIRST ELEMENT OF SS:

S("S1") = YES;

*       ONE ELEMENT OF SET S IS GENERATED FOR EACH COMBINATION OF PURE STRATEGIES:

LOOP((I1,I2,I3),

*       SET S CONTAINS A SINGLE ELEMENT FOR THE FOLLOWING STATEMENT:

        LINK(S,I1,"J1") = YES;
        LINK(S,I2,"J2") = YES;
        LINK(S,I3,"J3") = YES;
        PAYOFF(S,J) = -LOSS(J,I1,I2,I3);

*       THE NEXT STATEMENTS MOVE S THE SUBSEQUENT ELEMENT OF SS:

        S(SS)$S(SS-1) = YES;
        S(SS)$S(SS+1) = NO;
);

*       THE NEXT STATEMENTS DEFINE THE SET OF ACTION PROFILES AS
*       THOSE ELEMENTS OF SS FOR WHICH PAYOFFS HAVE BEEN ASSIGNED:

S(SS) = NO;   S(SS) = YES$SUM(J, ABS(PAYOFF(SS,J)));
```

```
POSITIVE
VARIABLE        P(I,J)          PROBABILITY OF STRATEGY I BY PLAYER J,
                Y(J)            SIMPLEX VARIABLE (=1 IN EQUILIBRIUM)
                PI(J)           EXPECTED RETURN TO PLAYER J;

EQUATION        PIDEF(J)        EXPECTED PAYOFF DEFINITION,
                SIMPLEX(J)      SIMPLEX CONSTRAINT
                EQUIL(I,J)      EXCESS RETURN FROM PURE STRATEGY;


*       EXPECTED PAYOFF:

PIDEF(J)..

   PI(J) =E= SUM(S, PAYOFF(S,J) * PROD(JJ, SUM(I$LINK(S,I,JJ), P(I,JJ)/Y(JJ))));


*       STRATEGY I BY PLAYER J YIELDS NO MORE THAN EXPECTED RETURN:

EQUIL(I,J)..

   PI(J) =G= SUM(S$LINK(S,I,J),  PAYOFF(S,J) *

          PROD(JJ$(ORD(JJ) NE ORD(J)), SUM(II$LINK(S,II,JJ), P(II,JJ)/Y(JJ))));


*       PROBABILITIES SUM TO 1:

SIMPLEX(J)..    SUM(I, P(I,J)) =G= 1;
```

Figure 7: GAMS/MCP Code for the General N-Player M-Strategy Nash Game

matrices is easy in theory but tedious in practice. Fortunately, there is an easier approach by which GAMS generates the derivatives automatically.

Consider the following nonlinear program:

$$\max_{z,t} z_i$$

$$\text{s.t.} \quad F(z;t) = 0$$

$$t = \bar{t}$$

The implicit function theorem (see Varian (1992)) implies that the ith row of S corresponds to the dual multipliers associated with the bounds on t. To compute the entire sensitivity matrix therefore requires the solution of N separate sets of nonlinear programs, all of which may be initiated at the equilibrium point.

A minor variation on this technique applies to complementarity problems involving mixtures of equations and inequalities. In the MCP case, the nonlinear system passed to the nonlinear program only include those equations which are *binding* at the solution (i.e., any equations which are slack in the MCP solution must be omitted from the NLP using GAMS exception ("$") operators). An illustration of this technique is provided in the Appendix.

5 GAMS/MCP Syntax

In a GAMS/MCP program, the central unknowns, z, are declared as VARIABLES, the vectors u and are specified as upper and lower bounds. The function $F()$ is written using GAMS matrix algebra in equation statements. All aspects of the GAMS/MCP language are identical to GAMS/NLP except that in an MCP problem no objective function is specified, and bounded variables must be mapped to complementary inequalities.

In the canonical MCP structure, the equations $w^T(z - \ell) = 0$ and $v^T(u - z) = 0$ imply a particular association of variables with equations. Quite naturally, z_i is complementary with $F_i(z)$. In an empirical model, there are typically several classes of variables and equations, and the declaration sequence is arbitrary, so the rule $z_i F_i = 0$ is impossible to interpret.[8]

To fix the complementarity structure, a GAMS/MCP program explicitly associates variables with equations in the MODEL statement. Unlike an NLP in which only equation names appear on the model statement, in an MCP program both equation and variable names may be listed. The syntax is:

```
MODEL  NAME /EQU1.VAR1, EQU2.VAR2, .../;
```

In this list, `EQU_i` and `VAR_i` must be defined over the same domains.

Complementarity associations are only required for variables which are subject to upper or lower bounds. Otherwise, GAMS/MCP only requires that the number of unrestricted variables equal the number of unassociated equations in the model. It is a simple matter to "mix" equations with inequalities in this format. The specification of linear and nonlinear systems of equations with no bounds and complementarity conditions requires no variable names on the MODEL list.

Joint consistency of inequalities and complements assignments is important when finite bounds are imposed on z. In the MCP structure, slack variables w and v are implicitly represented

[8] One way to make this association would have been to use the same name for both equations and variables. This syntax was rejected because it would not be "upward compatible" with NLP models. The present structure permits "sharing" a system of equations in both complementarity and optimization models within the same program. Furthermore, the present structure makes it quite simple to convert existing GAMS/NLP models into the complementarity structure.

	Associated Equation Type		
Variable	=G=	=L=	=E=
Lower	OK	Error	OK/warning
Upper	Error	OK	OK/warning
Both	OK/warning	OK/warning	OK/warning
Free	OK/message	OK/message	OK
Fixed	OK	OK	OK

```
Key:

* ``Error'' means that the correspondence is inconsistent and the
problem will not be processed.

* ``OK/warning'' means that the association is acceptable but the
constraint as written may not be satisfied in the solution.  We
accept the syntax and issue a warning on the list file if the
constraint is not satisfied at the equilibrium.  This warning is
generated if the constraint is violated either by non-zero upper and
lower bound slack variables or through the effects of the multipliers
from the variational inequality side constraints.

* ``OK/message'' means that an inequality constraint has been
associated with an unrestricted variables.  The variable will always
be in the basis, so the constraint will never be slack and it is
therefore treated as an equality.  A message to this effect will be
generated on the listing file.
```

Figure 8: Rules for Associating Complementary Variables and Equations

in a GAMS/MCP model through the equation statement. The fact that the lower bound slack variable (w) appears with a coefficient of -1 and the upper bound slack variable (v) appears with a coefficient of $+1$ means that the "orientation" of the inequality is significant. In GAMS/MCP, the equations "$F(z) \geq 0$" and "$-F(z) \leq 0$" are *not* equivalent. In the first case, the inequality is assumed to be complementary with the lower bound slack of the associated variable while in the second case the inequality is taken to be complementary with the upper bound slack.

Compiler error conditions can result from improperly specified bounds and associated equation types. If an equation is written using `=G=` then the associated lower bound must be finite. Likewise, if a constraint is written as `=L=` the associated upper bound must be finite. Table A.1 outlines the rules which apply when variables are associated with different types of equations.

6 References

ADELMAN, I. and S. ROBINSON (1978) *Income Distribution Policy in Developing Countries: A Case Study of Korea*, Oxford Univeristy Press, London.

ANSTREICHER, K, J. LEE and T.F. RUTHERFORD (1992) "Crashing a Maximum Weight Complementary Basis", *Mathematical Programming* 54(3), 281-290.

BROOKE, T., D. KENDRICK and A. MEERAUS (1987) *GAMS: A User's Guide*, Scientific

Press, South San Francisco.

DAFERMOS, S. (1983) “An Iterative Scheme for Variational Inequalities”, *Mathematical Programming* 26, 40-47.

DANTZIG, G. (1963) *Linear Programming and Extensions* Princeton University Press.

DIRKSE, S. and M. FERRIS (1993) “The PATH Solver: A Non-Monotone Stabilization Scheme for Mixed Complementarity Problems”, Tech Report 1179, CS Dept, UW-Madison, September.

DUFFIE, D. and W. SHAFER (1985) “Equilibrium in Incomplete Markets: I. A Basic Model of Generic Existence”, *Journal of Mathematical Economics* 14, 285-300.

EAVES, B.C. (1978) “A locally quadratically convergent algorithm for computing stationary points,” Tech. Rep., Department of Operations Research, Stanford University, Stanford, CA.

GALE, D. (1960) *The Theory of Linear Economic Models*, University of Chicago Press.

GILL, P., W. MURRAY, M.A. SAUNDERS and M.H. WRIGHT (1991) “Maintaining LU factors of a general sparse matrix, *Linear Algebra and its Applications* 88/89, 239-270.

GOREUX, L.M. and A.S. MANNE (1973) *Multi-level Planning: Case Studies in Mexico*. Amsterdam: North-Holland.

HARKER, P. (1986) “Alternative Models of Spatial Competition”, *Operations Research*, 34(3), 410-425.

HARKER, P. and J.S. PANG (1990) “Finite-Dimensional Variational Inequality and Nonlinear Complementarity Problems: A Survey of Theory, Algorithms and Applications”, *Mathematical Programming B*, 38(2), 161-190.

HOGAN, W.W. (1975) “Energy policy models for project independence,” *Computation and Operations Research* 2, 251-271.

JOSEPHY, N.H. (1979) “Newton’s method for generalized equations”, Techncical Summary Report #1965, Mathematical Research Center, University of Wisconsin - Madison.

KOJIMA, M., N. MEGIDDO, T. NOMA and A.YOSHISE (1991) “A unified approach to interior point algorithms for linear complementarity problems”, *Lecture Notes in Computer Science* 538, Springer-Verlag.

LAAN, G. van der, and A.J.J. TALMAN (1985) “An algorithm for the linear complementarity problem with upper and lower bounds”, *Journal of Optimization Theory and Applications* 62(1), 151-163.

LAAN, G. van der, and A.J.J. TALMAN and L. van der HEYDEN (1987) “Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling”, *Mathematics of Operations Research* 12(3), 337-397.

LEMKE, C.E. and J.T. HOWSON (1964) “Equilibrium points of bimatrix games,” SIAM Review 12, 413-23.

MATHIESEN, L. (1985) “Computation of Economic Equilibrium by a Sequence of Linear Complementarity Problems”, *Mathematical Programming Study 23*, North-Holland.

MEERAUS, A. (1983) “An algebraic approach to modeling,” *Journal of Economic Dynamics and Control*, 5(1), February.

RALPH, D. (1994) “Global Convergence of Damped Newton’s Method for Nonsmooth Equations, via the Path Search”, *Mathematics of Operations Research*, 19(2), 352-371.

ROBINSON, S. (1972) “A quadratically-convergent algorithm for general nonlinear programming problems,” *Mathematical Programming Study* 3, 145-156.

RUTHERFORD, T. (1987) “Applied general equilibrium modeling”, Ph.D. Dissertation, Department of Operations Research, Stanford University.

RUTHERFORD, T. (1992a) “Sequential joint maximization”, Working Paper 92-8, Department of Economics, University of Colorado.

RUTHERFORD, T. (1992b) “Applied general equilibrium modeling using MPS/GE as a GAMS subsystem”, Working Paper 92-15, Department of Economics, University of Colorado.

RUTHERFORD, T. (1993) “MILES: a Mixed Inequality and nonLinear Equation Solver”, Working Paper, Department of Economics, University of Colorado.

SAMUELSON, P. (1952) “Spatial price equilibrium and linear programming,” *The American Economic Review*, 42, 283-303.

TAKAYAMA, T. and G.G. JUDGE (1971) *Spatial and Temporal Price and Allocation Models*, North Holland, Amsterdam.

VARIAN, H. (1992) *Microeconomic Analysis (3rd edition)*, Norton, New York.

Mixed Complementarity Programming with GAMS

T.F. Rutherford (JECD 1998)

Update section 3: Transport problem with isoelastic supply and demand functions

Instead of :

$$\gamma_i = \frac{\eta_i}{1+\eta_i}\left(\frac{1}{\alpha_i}\right)^{1/\eta_i} \text{ and } \quad \gamma_i = \frac{\sigma_j}{1+\sigma_j}\beta_j^{1/\sigma_i}$$

it should be:

$$\gamma_i = \frac{1}{1+\eta_i}\left(\frac{1}{\alpha_i}\right)^{1/\eta_i} \text{ and } \quad \gamma_i = \frac{1}{1-\sigma_j}\beta_j^{1/\sigma_j}$$

Also: In the objective function, the signs on the integrals of demand and supply functions should be opposite.