A GLOBAL NEWTON METHOD TO COMPUTE NASH EQUILIBRIA

SRIHARI GOVINDAN AND ROBERT WILSON

Abstract. A new algorithm is presented for computing Nash equilibria of finite games. Using Kohlberg and Mertens' structure theorem we show that a homotopy method can be represented as a dynamical system and implemented by Smale's global Newton method. The algorithm is outlined and computational experience is reported. JEL Classification C63.

1. Introduction

Two global path-following algorithms for solving systems of equations are the global Newton method of Smale (1976) and the homotopy method of Eaves (1972, 1984). Here we combine these methods to compute Nash equilibria of finite games. The feature that facilitates our construction is a fundamental topological property of the graph of the Nash equilibrium correspondence. The structure theorem of Kohlberg and Mertens (1986, Theorem 1) shows that this graph is homeomorphic to the space of games. The homeomorphism enables a homotopy method to be applied in a subspace of games whose dimension is the same as the number of pure strategies. The path of the homotopy can be traced by a dynamical system using the global Newton method; then afterwards one applies the inverse of the homeomorphism to obtain the players' mixed strategies at the equilibrium.

The resulting algorithm subsumes many others as special cases, including linear complementarity algorithms for two-player games (Lemke and Howson, 1964; Lemke, 1965; Eaves, 1971; Tomlin, 1978; Shapley, 1974) and many-player polymatrix games (Howson and Rosenthal, 1974), extensions to perfect, simply-stable, and proper equilibria (Wilson, 1992; Yamamoto, 1993), nonlinear extensions to many-player games (Rosenmüller, 1971; Wilson, 1971), and applications to Walrasian equilibria of economic models (Balasko, 1988; Eaves and Schmedders, 1999, and their extensive references; Scarf and Hansen, 1973; Wilson, 1978). It differs fundamentally from the differentiable homotopy algorithm of Herings and Peeters (2001) that implements Harsanyi and Selten's (1988) linear tracing procedure in the strategy space along a path that deforms an arbitrary initial strategy into an equilibrium strategy.

Our formulation is motivated initially by the homotopy principle (Eaves, 1972, 1984; Eaves and Schmedders, 1999): given a system of equations whose zeros one wants to compute, first deform the system to one with a unique easily-computed solution and then reverse the deformation and trace solutions of the associated systems along the path to find a solution of the original system at the terminus. Here we solve a system stating the conditions for an equilibrium of a specified game, so the homotopy method involves reversing a deformation of the true game into another game with a unique equilibrium. A key issue is how to exploit the game-theoretic aspects to best advantage. The structure theorem provides an answer: one of its implications is that, above each generic ray emanating from the true game (represented as a point in a Euclidean space),
the graph of the equilibrium correspondence is a 1-dimensional manifold; further, at a sufficient distance from the true game there is a unique equilibrium. Therefore, starting from a sufficiently distant game along any generic ray, one can traverse the line segment to the true game, tracing the 1-dimensional manifold of equilibria along the way, to find an equilibrium of the true game at the terminus.

For practical computation, however, one must eliminate the huge discrepancy between the dimensions of the spaces of games and mixed strategies. The structure theorem again provides an answer: the homeomorphism shows that it is sufficient to consider a subspace of deformed games of the same dimension as the space of mixed strategies, and it provides a simple parameterization. Indeed, it shows that one can represent both deformed games and their equilibria within this subspace, deferring until last the calculation of the true game’s equilibrium strategies by applying the homeomorphism. But now, since the two dimensions are the same, the global Newton method can be used as the engine for calculations. All this leads to a formulation in which one traces the 1-dimensional manifold above a generic ray (emanating from the true game positioned at the origin) via the dynamical system of the global Newton method. The result is an explicit algorithm that accomplishes the reverse deformation suggested by the homotopy principle.

Our analysis establishes the conditions for application of the homotopy method and then shows how it can be implemented by the global Newton method [GNM]. We follow the usual convention that GNM is formulated as a dynamical system with a continuous time parameter. The theoretical simplicity comes at some cost because the technical issue of differentiability must be addressed. The standard formulation of GNM requires a $C^2$ map but in its application to games the inequality constraints on equilibria imply that one must contend with a map that is piecewise-$C^2$. We show in Sections 3 and 4 that this problem is minor because the qualitative aspects of the dynamical system depend only on the topological structure of the Nash graph. Formally, we construct a dynamical system that involves restarts. A byproduct of our approach is a generalization to many-player games of Shapley’s (1974) demonstration that the index is always +1 for an equilibrium at a terminal point of the Lemke-Howson algorithm for generic 2-player games. In our case a terminal point of GNM has index +1, while equilibria with index −1 are locally unstable and obtained by GNM with the time direction reversed.

Section 2 reviews the structure theorem and Section 3 derives its implications for constructing GNM’s dynamical system, which is specified in Section 4. Section 5 describes an implementation of GNM and computational experience.

2. The Structure Theorem

Kohlberg and Mertens (1986) prove that the graph of the Nash equilibrium correspondence is homeomorphic to the space of games in normal form, with fixed finite sets of players and pure strategies. We describe the homeomorphism in this section and develop relevant implications.

Let $N$ be the finite set of players. For each player $n \in N$ let $S_n$ be $n$’s finite set of pure strategies and let $\Sigma_n$ be $n$’s simplex of mixed strategies on $S_n$. For each player $n$ denote by $S_{-n}$ the set $\prod_{i \neq n} S_i$ of pure-strategy profiles of $n$’s opponents, and define $S = \prod_{n \in N} S_n$, the set of pure-strategy profiles. The set $\Sigma = \prod_{n \in N} \Sigma_n$ of mixed-strategy profiles lies in the Euclidean space of dimension $m = \sum_n |S_n|$ and is a cell of dimension $m - |N|$. The space of games with these strategy sets is a Euclidean space $\Gamma$ of dimension $|N| \times \prod_{n \in N} |S_n|$ in which each point assigns to each of the $|N|$ players a payoff at each of the $|S|$ profiles of their pure strategies. Denote by $E$ the graph of the Nash equilibrium correspondence; that is, $E = \{(G, \sigma) \in \Gamma \times \Sigma$
\[ \text{ \sigma \ is \ an \ equilibrium \ of \ the \ game \ G} \}. \text{ Let } \bar{E} \text{ and } \bar{\Gamma} \text{ be \ the \ one-point \ compactifications \ (at } \infty \text{) \ of } E \text{ and } \Gamma. \text{ Use } p : E \to \Gamma \text{ to denote \ the \ natural \ projection \ from \ the \ graph \ to \ the \ space \ of \ games, \ and \ let } \bar{p} \text{ be \ the \ continuous \ extension \ of } p \text{ that \ maps } \infty \to \infty. \text{ By \ construction, \ the \ extended \ space } \bar{\Gamma} \text{ is \ a \ topological \ sphere \ of \ dimension } |N| \times |S|, \text{ and \ the \ structure \ theorem \ says \ that \ the \ extended \ graph } \bar{E} \text{ is \ homeomorphic \ to \ this \ sphere.} \]

\textbf{Theorem 2.1.} [Kohlberg and Mertens, 1986, Theorem 1] \text{ There \ exists \ a \ homeomorphism } H : E \to \Gamma. \text{ Further, } h = p \circ H^{-1} : \Gamma \to \Gamma \text{ is \ linearly \ homotopic \ to \ the \ identity \ map \ on } \Gamma, \text{ and \ this \ homotopy \ extends \ to } \bar{\Gamma}. \]

The existence of the homotopy ensures that the composite map } p \circ H^{-1} \text{ from } \bar{\Gamma} \text{ to itself has degree } +1 \text{ and that the extended graph } \bar{E} \text{ can be oriented so that the degrees of } H, h, \text{ and } p \text{ are each } +1. \text{ We omit the proof \ but \ describe \ the \ homeomorphism } H. \]

\text{Associate \ with \ each \ game } \bar{G} \text{ the } m \text{-dimensional vector } g \text{ for which } g_s \text{ is \ the \ expected \ payoff \ to \ player } n \text{ from strategy } s \in S_n \text{ when each other player mixes uniformly over his pure strategies. Then \ the \ space \ of \ games \ can \ be \ parameterized \ by \ decomposing \ each \ game } G \text{ into \ the \ pair } (\bar{G}, g) \text{ where } \sum_{t \in S_{-n}} \bar{G}_{s,t} = 0 \text{ for \ each } s \in S_n. \text{ In \ this \ representation \ the \ payoff \ to } n \text{'s \ pure \ strategy } s \text{ when \ other \ players \ use \ the \ mixed \ strategy } \sigma \text{ is} \]

\[ v_s(\sigma) = \sum_{t \in S_{-n}} \bar{G}_{s,t} \prod_{i \neq n} \sigma_i = g_s + \sum_{t \in S_{-n}} \bar{G}_{s,t} \prod_{i \neq n} \sigma_i \]

\text{and \ player } n \text{'s \ payoff \ from \ an \ optimal \ reply \ is } \max_{s \in S_n} v_s(\sigma). \text{ The \ map } H : E \to \Gamma \text{ is \ specified \ by } H(\bar{G}, g, \sigma) = (\bar{G}, z), \text{ where } z_s = \sigma_s + v_s(\sigma) \text{ for \ each \ pure \ strategy } s \in S_n \text{ of \ player } n. \]

\text{A \ useful \ tool \ is \ the \ retraction \ map } r : \mathbb{R}^m \to \Sigma \text{ introduced \ by \ Gil\l, \ Pearce, \ and \ Stacchetti \ (1993). \ For \ each } z \in \mathbb{R}^m \text{ and } n \in N, \text{ let } v_n(z) \text{ be \ the \ real \ number \ that \ solves } \sum_{s \in S_n} (z_s - v_n(z))^+ = 1, \text{ where } (x)^+ = \max\{0, x\}. \text{ Then } r(z) = \sigma \text{ where } \sigma_s = (z_s - v_n(z))^+ \text{ for \ each } s \in S_n \text{ and } n \in N. \text{ Equivalently, } \sigma \text{ is \ the \ point \ in } \Sigma \text{ nearest \ to } z \text{ in \ Euclidean \ distance. \ Note \ that \ the \ retraction \ map } r \text{ \ depends \ only \ on \ the \ dimensions \ of \ the \ players’ \ normal-form \ strategy \ sets, \ independently \ of \ the \ payoffs } (\bar{G}, g). \]

\text{Using \ the \ retraction \ map } r, \text{ one \ computes \ the \ inverse \ image } H^{-1}(\bar{G}, z) \text{ as \ the \ point } (\bar{G}, g, \sigma) \text{ in } E \text{ for \ which } \sigma = r(z) \text{ and} \]

\[ g_s = z_s - \sigma_s - \sum_{t \in S_{-n}} \bar{G}_{s,t} \prod_{i \neq n} \sigma_i \]

\text{for \ each \ pure \ strategy } s \in S_n \text{ of } n. \text{ The \ gist \ of \ this \ construction \ is \ that } H \text{ \ augments \ each \ pure \ strategy’s \ probability \ by \ its \ expected \ payoff, \ which \ is \ reversible \ because \ the \ maximum \ payoff \ for \ a \ player \ can \ be \ recovered \ from \ that \ player’s } z \text{-values \ via \ the \ retraction } r. \text{ This \ corresponds \ to \ the \ definition \ of \ a \ Nash \ equilibrium \ that \ the \ pure \ strategies \ used \ with \ positive \ probability \ by \ a \ player \ must \ be \ among \ those \ with \ the \ maximum \ expected \ payoff.} \]

\text{For \ each \ fixed \ matrix } \bar{G}, \text{ let } \Gamma(\bar{G}) = \{(\bar{G}, g) \mid g \in \mathbb{R}^m\} \text{ be \ the } m \text{-dimensional \ space \ of \ games \ parametrized \ by } g \in \mathbb{R}^m. \text{ Then } H(\bar{G}, \cdot) \text{ specifies \ a \ homeomorphism \ onto } \Gamma(\bar{G}). \text{ We \ work \ henceforth \ with \ this \ restricted \ map \ for \ some \ fixed } \bar{G}, \text{ \ still \ calling \ it } H. \text{ Because } \Gamma(\bar{G}) \text{ can \ be \ identified \ with \ the \ space } \mathbb{R}^m, \text{ \ we \ view } r \text{ \ as \ a \ retraction \ of } \Gamma(\bar{G}) \text{ onto } \Sigma. \text{ The \ map } h : \Gamma(\bar{G}) \to \Gamma(\bar{G}) \text{ assigns \ to \ each } z \text{ \ the \ vector } g \text{ \ of \ average \ payoffs \ that}
makes the mixed-strategy profile \( r(z) \) an equilibrium of the game \((\hat{G}, g)\); that is, \( h(z) = z - r(z) - \hat{G}(r(z)) \)
where \( \hat{G}(\sigma)_s \) is the last term in the formula for \( v_s(\sigma) \) displayed above.

Figure 1 shows the relations among the maps mentioned above and the variables in their domains and images, using \( q : E \to \Sigma \) to indicate the projection to the strategy space.

Güл, Pearce, and Stacchetti (1993) use the retraction \( r \) to characterize an equilibrium as a fixed point \( \sigma = r(\hat{H}(\sigma)) \) of the composite function \( r \circ \hat{H} : \Sigma \to \Sigma \) using \( \hat{H}_s(\sigma) = \sigma_s + v_s(\sigma) \) when the game \( G \) is fixed. In our setup it is better to construct a fixed point \( z \) of the commuted function \( \hat{H} \circ r : \Gamma(\hat{G}) \to \Gamma(\hat{G}) \) and then afterwards to obtain the equilibrium as \( \sigma = r(z) \). (“Better” here refers not just to theoretical convenience but to our conclusion from numerical testing.)

The map \( H \) appears smooth but its domain is the Nash graph, which is not. The inverse map \( H^{-1} \) is piecewise-smooth in the usual sense. Specifically, for each \( T = \prod_n T_n \subseteq S \) the restriction of \( H^{-1} \), and hence also \( h \), to the set of those \( z \) such that the support of \( r(z) \) is \( T \) is a polynomial function of degree \(|N| - 1\) in \( z \). A generic game is a regular value of each of these polynomials. In particular, a generic normal-form game has a finite number of equilibria (Govindan and Wilson, 2001b).

The above properties of \( h \) yield a definition of the degree of an equilibrium, defined as the local degree of the projection \( p \) from the Nash graph to the space of games. This definition developed in Govindan and Wilson (1997) can be abbreviated here by using the composition \( p = h \circ H \). Because \( H \) is a homeomorphism, its local degree is \(+1\) so the local degree of \( p \) is the same as the local degree of \( h \). Call a game \( g \) generic if \( h \) is differentiable at every \( z \in h^{-1}(g) \) and the determinant of the Jacobian \( Dh \) is nonzero at \( z \). Then the Inverse Function Theorem implies that such a game \( g \) has a finite number of equilibria. Further, the local degree \( \text{Deg}(\sigma) \) of an isolated equilibrium \( \sigma \) is the sign of the determinant of the Jacobian \( Dh \) at \( z = H(g, \sigma) \) (Dold, 1972). The extension of \( h \) to \( \Gamma \) is homotopic to the identity so its global degree is \(+1\); consequently \( \sum_{z \in h^{-1}(g)} \text{Deg}(r(z)) = +1 \) for each generic game \( g \). The concept of degree extends to a component of equilibria in a simple way: choose an open neighbourhood of the component such that for each nearby game all equilibria are either in the neighbourhood or outside its closure; then the degree of the component is the sum of the degrees of all equilibria in the chosen neighbourhood of the component for any nearby generic game. Govindan and Wilson (1997) show that the degree of a component is the same as its index as usually defined for fixed points (Güл, Pearce, and Stacchetti, 1993).

3. Analytical Foundations

The global Newton method is motivated by Hirsh’s (1963) elegant proof that a ball \( B \) cannot be retracted onto its boundary \( \partial B \). Its simplest form is the contradiction obtained by supposing there is a smooth retraction \( R : B \to \partial B \). Then Sard’s theorem implies the existence of a regular value \( b \in \partial B \) for which \( R^{-1}(b) \) is a 1-dimensional manifold with boundary \( R^{-1}(b) \cap \partial B \). One connected component of \( R^{-1}(b) \) has \( b \) as one boundary point but because \( R \) is the identity on \( \partial B \), it cannot contain another boundary point.

This idea applies to computing rest points of vector fields. Let \( \phi \) be a continuous vector field on a ball \( B \) such that \( \phi(b) \) points inward at each boundary point \( b \in \partial B \), namely \( b \cdot \phi(b) < 0 \), and the differential equation defined by \( \phi \) has a well-defined solution for each initial condition \( b \in \partial B \). For each generic initial condition \( b \in \partial B \) the induced trajectory is a 1-dimensional manifold with \( b \) its only point in \( \partial B \); therefore every limit point of the trajectory is a rest point \( b^* \in \phi^{-1}(0) \).
In this and the next Section we develop an analogous procedure to construct a trajectory in the space of games whose limit point is an equilibrium of the game at the terminus. We consider a given game $G^* = (\tilde{G}^*, g^*)$, not necessarily generic, whose equilibria one wants to compute. For simplicity, we refer to this game as $g^*$ and we use the subspace $\Gamma(\tilde{G}^*)$ generated by varying only the parameter $g$ — recall from Section 2 that we identify $\Gamma(\tilde{G}^*)$ with $\mathbb{R}^m$.

The central object of our analysis is the map $\psi : \mathbb{R}^m \to \mathbb{R}^m$ defined by $\psi(z) = h(z) - g^*$; that is,

$$\psi_s(z) = z_s - \sigma_s - \sum_{t \in S_{-n}} \tilde{G}_{st}^* \prod_{i \neq n} \sigma_i - g_s^*$$

for each $n$ and $s \in S_n$, where $\sigma = r(z)$ is the retraction of $z$ to $\Sigma$. Note that $-\psi(z)$ is the displacement between the true game $g^*$ and the game $g = h(z)$ for which $\sigma = r(z)$ is an equilibrium; thus, $-\psi$ plays a role analogous to the vector field $\phi$ mentioned above. Let $E^* = \{(g^*, \sigma) \in E\}$ be the set of equilibria of the game $g^*$ viewed as a subset of the equilibrium graph $E$, and let $Z^* = H(E^*)$ be its image under the homeomorphism. Then $Z^* = \psi^{-1}(0)$ and $E^* = (g^*, r(Z^*))$. Thus the set $Z^*$ of zeros of $\psi$ identifies the set of the Nash equilibria of the game $g^*$.

The algorithm in Section 5 uses the global Newton method (Smale, 1976; Hirsch and Smale, 1979) to compute one or more zeros in $Z^*$ of the function $\psi$. After having obtained a zero $z^*$ one applies the retraction $r$ to obtain the corresponding equilibrium $\sigma^* = r(z^*)$ of the game $g^*$. The objective of this Section is to show that the map $\psi$ is sufficiently well behaved that the global Newton method can be applied to find a zero.

Denote by $S$ the unit sphere in $\mathbb{R}^m$, and define the map $f : \mathbb{R}^m \setminus Z^* \to S$ by $f(z) = \psi(z)/\|\psi(z)\|$. For every $b \in S$, $f^{-1}(b)$ is the image under $H$ of $\{(g^* + \lambda b, \sigma) \in E \mid \lambda > 0\}$; i.e., $f^{-1}(b)$ can be viewed as the section of the graph of equilibria above the ray $g^* + \lambda b$. The result of this Section is the following Proposition describing the set $f^{-1}(b)$ for a generic $b \in S$.

**Proposition 3.1.** There exists a closed, semi-algebraic, lower-dimensional subset $C$ of $S$ such that for each point $b \in S \setminus C$:

1. $f^{-1}(b)$ is a boundaryless, 1-dimensional, semi-algebraic manifold that is a finite union of analytic manifolds.
2. The closure of each connected component of $f^{-1}(b)$ is a 1-manifold whose boundary is included in $Z^*$.
3. There exists $\lambda > 0$ such that for each $\lambda \geq \lambda$, the game $g^* + \lambda b$ has a unique equilibrium, which is a pure-strategy profile. There exists a unique component of $f^{-1}(b)$ that contains the $H$-images of these equilibria. This component has a unique boundary point, which is a zero in $Z^*$.

Before proving Proposition 3.1, we illustrate the topological possibilities it implies. Figure 2 depicts schematically a 1-dimensional cross-section of the equilibrium graph; in general the lines are curved but we omit this feature. Viewing the true game $g^*$ as located at the origin, Figure 2 shows the equilibrium graph over two opposite rays, one through $b \in S \setminus C$ and the other through $-b \in S \setminus C$. The game $g^*$ has three equilibria, two with index $+1$ and one with index $-1$. Over the ray through $b$, there are three 1-dimensional manifolds. One of them is a compact manifold without boundary (a cycle) while the other two have equilibria of $g^*$ as their boundary points. Proposition 3.1 shows that (the homeomorphic images of) these are the only possible types of manifolds that can occur. Moreover, along each ray each game sufficiently far from the origin has a unique equilibrium, and these games and their equilibria belong to a component that has an
equilibrium of \( g^* \) whose index is +1 because the game is generic. Our proof of Proposition 3.1 actually describes the analytic manifolds asserted in Statement 1.

First we introduce notation that is used again in Section 4. Given \( z \in \mathbb{R}^m \) and \( n \in N \), let \( v_n(z) \) be the equilibrium payoff function defined in Section 2, i.e., \( v_n(z) \) is the unique real number for which \( \sum_{s \in S_n} (z_{n,s} - v_n(z))^+ = 1 \). For each support \( T = \Pi_n T_n \subseteq S \), let \( Z(T) \) be the set of \( z \in \mathbb{R}^m \) for which: (i) \( z_{n,s} \geq v_n(z) \) for all \( n \in N, s \in T_n \); (ii) \( z_{n,s} \leq v_n(z) \) for all \( n \in N, s \notin T_n \); and (iii) \( z_{n,s} = v_n(z) \) for at most one \( n \in N, s \in S_n \). \( Z(T) \) is a linear \( m \)-manifold with boundary \( \partial Z(T) \) given by those points \( z \) for which \( z_{n,s} = v_n(z) \) for exactly one \( n, s \). The intersection of two sets, say \( Z(T) \) and \( Z(T') \), when nonempty is a common “face” of both sets; and every face of the boundary of \( Z(T) \) belongs to exactly one of the other sets. Observe that \( H^{-1}(Z(T)) \) is the set of points \( (g, \sigma) \in E \) such that: (i) for all \( n \in N \) and \( s \in T_n \), \( s \) is a best reply to \( \sigma \) in the game \( g \); (ii) the support of \( \sigma \) is contained in \( T \); and (iii) there exists at most one \( n \in N \) and \( s \in S_n \) such that \( s \) is a best reply against \( \sigma \) in \( g \) but is used with zero probability in \( \sigma \).

**Proof of Proposition 3.1.** For each support \( T \), the restriction of \( f \) to \( Z(T) \setminus Z^* \) is an analytic semi-algebraic function. Therefore, there exists a lower-dimensional subset \( C(T) \) of \( S \) such that each \( b \in S \setminus C(T) \) is a regular value of the restrictions of \( f \) to \( Z(T) \) and \( \partial Z(T) \). In particular, for each such \( b \), \( f^{-1}(b) \) is either empty or a 1-dimensional semi-algebraic (analytic) manifold with boundary equal to its intersection with \( \partial Z(T) \). Let \( C_0 = \cup_{T} C(T) \) be the union over all supports of the critical sets \( C(T) \). Because \( (Z \setminus \cup_T Z(T)) \setminus Z^* \) is a set of dimension \( m - 2 \), its image under \( f \), call it \( C_1 \), is a lower dimensional subset of \( S \). Let \( C' = C_0 \cup C_1 \).

For \( b \in S \setminus C' \), \( f^{-1}(b) \subset \cup_T Z(T) \), and moreover, its intersection with each \( Z(T) \) is either empty or a 1-dimensional semi-algebraic manifold. Hence \( f^{-1}(b) \) is a 1-dimensional semi-algebraic set that is a finite union of analytic manifolds. To prove that \( f^{-1}(b) \) is itself a manifold, we show that it contains no branch point. Obviously, a point in the interior of some \( Z(T) \) is not a branch point, since each such point has a neighbourhood in \( f^{-1}(b) \) that is homeomorphic to an interval. Likewise, a point in the boundary of some \( Z(T) \) belongs to the boundary of exactly one other such set, and then it has in each of these two sets a neighbourhood that is homeomorphic to a half-closed interval, implying readily that it cannot be a branch point. Therefore, \( f^{-1}(b) \) is a boundaryless 1-dimensional semi-algebraic manifold. This proves Statement 1.

To prove Statement 2, observe that each connected component \( M \) of \( f^{-1}(b) \) is a semi-algebraic 1-dimensional manifold. Hence its closure \( \overline{M} \) can be triangulated as a 1-dimensional simplicial complex. It follows readily that \( \overline{M} \) is also a 1-manifold, since it is obtained from \( M \) by adding vertices. Finally, a boundary point of \( M \) cannot belong to \( f^{-1}(b) \) and therefore belongs to \( Z^* \).

Now Statement 3. There exists a lower-dimensional set \( C_2 \) of \( S \) such that for each \( b \in S \setminus C_2 \), each player \( n \) has a unique pure strategy \( \bar{s}_n \in S_n \) such that \( b_{\bar{s}_n} = \max_{s \in S_n} b_s \). When \( \lambda \) is sufficiently large, the pure strategy \( \bar{s}_n \) of each player \( n \) strictly dominates all others in the game \( g^* + \lambda b \), which therefore has a unique Nash equilibrium \( \bar{s} \equiv (\bar{s}_n)_{n \in N} \). Let \( C = C' \cup C_2 \). For each \( b \in S \setminus C \), there exists \( \lambda > 0 \) such that for each \( \lambda \geq \overline{\lambda} \) the game \( g^* + \lambda b \) has the unique equilibrium \( \bar{s} \). Therefore, there is a unique component \( M \) of \( f^{-1}(b) \) that contains the \( H \)-image of these equilibria. By Statement 2, the closure \( \overline{M} \) of \( M \) is a manifold. Let \( M^o = M \cap \psi^{-1}(\{ \lambda b : \lambda > \overline{\lambda} \}) \). \( \overline{M} \setminus M^o \) is a closed connected manifold because \( M^o \) is homeomorphic to an open interval and the games \( g^* + \lambda b \) for \( \lambda > \overline{\lambda} \) have the same unique equilibrium \( \bar{s} \). Moreover, \( \overline{M} \setminus M^o \) is a subset of \( \psi^{-1}(\{ \lambda b : 0 \leq \lambda \leq \overline{\lambda} \}) \), which is compact. Therefore \( \overline{M} \setminus M^o \) is a compact connected manifold with boundary. One of its boundary points is the \( H \)-image of the unique pure-strategy equilibrium \( \bar{s} \) of the game \( g^* + \lambda b \). Its other boundary point is obviously a zero in \( Z^* \). \( \square \)
We conclude this Section with two remarks about Proposition 3.1. (i) One might be tempted to conclude that the closure of \( f^{-1}(b) \) is itself a manifold with boundary for generic \( b \). This statement is true if the game \( g^* \) is itself generic, but without this assumption a zero in \( Z^* \) for the game \( g^* \) might be a branch point of the closure of \( f^{-1}(b) \). (ii) While Statements 1 and 2 are valid in the full space \( \Gamma \), Statement 3 is not. For instance, let \( G^* = (0,0) \) be the trivial 2-player 2-by-2 game and let \( G = (I,I) \) specify the ray; then \( G(\lambda) = G^* + \lambda G \) is the coordination game

\[
\begin{pmatrix}
\lambda, \lambda & 0, 0 \\
0, 0 & \lambda, \lambda
\end{pmatrix}.
\]

For each \( \lambda > 0 \) the game \( G(\lambda) \) has three equilibria, and this situation is generic because the same holds for all games near \( G(\lambda) \).

4. GNMs Dynamical System

The objective of this Section is to adapt Smale’s global Newton method to tracing a path in the 1-dimensional manifold \( f^{-1}(b) \). An outline is as follows. In subsection 4.1 we describe a particular orientation of the manifold \( f^{-1}(b) \) using the standard construction in Hirsch and Smale (1979). Then in 4.2 we show that the orientation, viewed as a dynamical system, induces a well-defined piecewise-analytic trajectory and we describe the limit point of the trajectory. In 4.3 we show that the dynamical system has well-behaved local stability properties when the game \( g^* \) is generic. Finally, in 4.4 we give a formula for the orientation that is the basis for the algorithm in Section 5.

We use the following notation in this section: for each support \( T = \prod_n T_n \subseteq S \), let \( (X(T), \partial X(T)) = (Z(T), \partial Z(T)) \cap f^{-1}(S \setminus C) \), where \( C \) is the critical set described in Proposition 3.1. Henceforth we consider only those supports \( T \) for which \( X(T) \) is nonempty. This innocuous convention excludes full-dimensional sets \( Z(T) \) mapped by \( f \) into the lower-dimensional critical subset \( C \) of \( S \).

4.1. Orienting the 1-manifold \( f^{-1}(b) \). For each support \( T \), define the orientation map \( \theta_T : X(T) \to \mathbb{R}^m \) as follows: for each \( z \in X(T) \), pick an ordered basis \( e_1(z), \ldots, e_{m-1}(z) \) of the normal space to \( \text{Ker} Df_z \) such that the collection \( f(z), [Df_z] \cdot e_1(z), \ldots, [Df_z] \cdot e_{m-1}(z) \) defines the positive orientation of \( \mathbb{R}^m \). Now define \( \theta_T(z) \) to be the unique unit vector in \( \text{Ker} Df_z \) such that the ordered basis \( \theta_T(z), e_1(z), \ldots, e_{m-1}(z) \) defines the negative orientation of \( \mathbb{R}^m \). This definition of \( \theta_T(z) \) is independent of the choice of the basis vectors; in fact, one could choose any set of \( m-1 \) vectors whose images under \( Df_z \) are linearly independent. By construction, \( \theta_T \) defines an orientation of the manifold \( f^{-1}(b) \cap X(T) \) for each \( b \), and one verifies easily that \( \theta_T \) is analytic. Moreover, if \( z \) belongs to \( \partial X(T) \) then, using the fact that \( z \) is a regular point of the restriction of \( f \) to \( \partial Z(T) \) because \( b \notin C \), \( \theta_T(z) \) cannot belong to the tangent space to \( \partial Z(T) \) at \( z \), i.e., it points either inwards or outwards. The following Proposition shows that the various maps \( \theta_T \) match up at the boundaries.

Proposition 4.1. Suppose \( z \in \partial X(T) \cap \partial X(T') \) for two different supports \( T \) and \( T' \). Then \( \theta_T \) points inward along \( \partial X(T) \) iff it points outward along \( \partial X(T') \).

Proof. Since \( z \) is a regular point of the restriction of \( f \) to \( \partial X(T) \), we can choose a basis \( w_1, \ldots, w_{m-1} \) for the tangent space \( T_z \partial X(T) \) at \( z \) such that \( f(z), [Df_z] \cdot w_1, \ldots, [Df_z] \cdot w_{m-1} \) defines the positive orientation of \( \mathbb{R}^m \). Then, by definition, both \( \theta_T(z), w_1, \ldots, w_{m-1} \) and \( \theta_{T'}(z), w_1, \ldots, w_{m-1} \) define the negative orientation.
of \(\mathbb{R}^m\). Therefore, both \(\theta_T\) and \(\theta_{T'}\) lie in \(\mathbb{R}^m\) on the same side of the linear subspace \(T_z\partial X(T)\), which implies the result.

\[\square\]

### 4.2. The flow \(\varphi\) generated by the orientations \(\theta_T\)

In this subsection we fix a generic \(b \in S \setminus C\) and a connected component \(M\) of \(f^{-1}(b)\).

For each fixed support \(T\) and \(z_0 \in X(T) \cap M\) consider the differential equation: \(\dot{z} = \theta_T(z);\; z(0) = z_0\). Since \(\theta_T\) is analytic, there exists a unique maximal (analytic) solution \(\varphi_{z_0} : I(z_0) \to M \cap X(T)\) where \(I(z_0)\) is an interval in the Reals containing zero and \(\varphi_{z_0}(0) = z_0\) and \(\frac{d\varphi_{z_0}}{dt}|_{t=0} = \theta_T(\varphi_{z_0}(s))\) for all \(s \in I\). Let \(\ell\) and \(\ell'\) be the two end points of \(I\). If \(\ell' \in I\) then \(\varphi_{z_0}(\ell)\) belongs to \(\partial X(T)\) and \(\dot{z}\) points outward there. In this case, there exists a unique \(T' \neq T\) such that \(z(\ell) = \theta_T(\ell)\) belongs to \(\partial X(T')\). Also, by Proposition 4.1, \(\theta_{T'}\) points inward at \(z(\ell)\). Therefore, the flow \(\varphi_{z_0}\) can be extended into \(X(T') \cap M\). If \(\ell'\) does not belong to \(I(z_0)\), then either \(\varphi_{z_0}(t)\) converges to a zero in \(Z^*\) or diverges to infinity. (We actually show below that the flow cannot diverge to infinity in forward time, only in backward time.) These arguments also apply for \(\ell\). This proves

**Theorem 4.2.** There exists an interval \(I \subseteq \mathbb{R}\) and a surjective map \(\varphi : I \to M\) such that: (i) \(\varphi\) is analytic outside a finite subset; (ii) for each \(s \in I\), if \(\varphi(s)\) belongs to \(X(T) \cap \partial X(T)\) for some support \(T\), then \(\frac{d\varphi(t)}{dt}|_{t=s} = \theta_T(\varphi(s))\).

The points where the flow \(\varphi\) is not analytic are precisely those in one of the boundaries \(\partial X(T)\). The schematic representation in Figure 2 includes arrows to indicate the orientation of the flow induced in the strategy space; i.e., the direction of motion as time increases. Also represented are folds and vertical segments in the graph that occur only at nongeneric games \(g = h(z)\) for \(z \in \partial X(T)\) where the support \(T\) changes.

We now provide a geometric description of the flow \(\varphi\). As in Smale (1976) one uses the Chain Rule to obtain

\[
Df_z(v) = \frac{D\psi_z(v)}{\|\psi(z)\|} - \frac{\psi(z)}{\|\psi(z)\|^3}(\psi(z) \cdot D\psi_z(v))
\]

at each \(z \in X(T)\) and \(v \in \mathbb{R}^m\). Because \(\theta_T(z) \in \text{Ker}Df_z\), there exists therefore a unique number \(\lambda(z) \in \mathbb{R}\) such that \([D\psi_z] \cdot \theta_T(z) = \lambda(z)\psi(z)\). The construction of the orientation implies that if \(D\psi_z\) is nonsingular then \(\lambda(z)\) and the determinant \(\text{Det}(D\psi_z)\) have opposite signs. Suppose \(\varphi\) is analytic at \(t \in I\) and let \(z = \varphi(t)\); then

\[
\frac{d\psi(z)}{dt} = [D\psi_z] \cdot \theta_T(z) = \lambda(z)\psi(z) \quad \text{and} \quad \frac{d\|\psi(z)\|}{dt} = \lambda(z)\|\psi(z)\|.
\]

Since \(\lambda(z)\) and \(\text{Det}(D\psi_z)\) have opposite signs, \(\|\psi\|\) decreases or increases as the determinant \(\text{Det}(D\psi_z)\) is positive or negative, and it remains constant where \(\text{Det}(D\psi_z) = 0\). Lifting the trajectory \(\varphi\) via \(H^{-1}\) to the equilibrium graph, yields the following geometric picture: the trajectory moves monotonically towards (resp., away from) the true game in sections of the graph where the degree of the (necessarily isolated) current equilibrium is \(+1\) (resp., \(-1\)). Since an analytic function that is zero on an open interval is zero everywhere, we also get a characterization of the regions where the trajectory moves vertically: for each \(T\) and each connected component of \(M \cap X(T)\), the flow is either vertical everywhere or at a finite number of points. We conclude this subsection with a characterization of the limit points of the flow \(\varphi\) (cf. Figure 2).

**Theorem 4.3.** The flow \(\varphi\) has the following properties:
1. If $M$ is a compact manifold, then $I = (-\infty, \infty)$; i.e., $\varphi : (-\infty, \infty) \to M$.
2. If $M$ is not compact but has a compact closure, then $I$ does not contain its end points and $\varphi(t)$ converges to a zero in $Z^*$ as $t$ converges to either end point of $I$.
3. If the closure of $M$ is not compact, then it is the unique manifold given by Statement 3 of Proposition 3.1. In this case, $I$ contains neither of its end points: $\varphi(t)$ diverges to $\infty$ as $t$ converges to the left end point, and converges to a zero in $Z^*$ as $t$ converges the right end point.

Proof. The proof of Statement 1 is standard; in this case $M$ is the image of a cycle. For Statements 2 and 3, observe that $M$ has a compact closure if $H^{-1}(M)$ does not contain equilibria of games sufficiently far along the ray $g^* + \lambda b$. Therefore, if $M$ is not compact and has a compact closure, then its boundary points are zeros in $Z^*$, which implies Statement 2. For Statement 3, suppose the closure of $M$ is not compact. Then, by Statement 3 of Proposition 3.1, there exists $\lambda > 0$ such that for each $\lambda \geq \overline{\lambda}$, the game $g^* + \lambda b$ has a unique equilibrium (in pure strategies) which $H$ then maps to $M$. Since the degree of a unique equilibrium is $+1$, $d||\psi(z)||/dt = \text{constant}$ is negative at every such point. Thus, starting at such a point, and running time backwards, we get divergence. Finally, $M$ contains a unique boundary point that is a zero in $Z^*$ to which $\varphi(t)$ must converge in forward time.

Starting from each initial point, the flow $\varphi$ generates a trajectory in $M \subset \mathbb{R}^m \approx \Gamma(G)$, the space of games with $G$ fixed. We call the image under $r$ of a trajectory, the induced trajectory in the strategy space $\Sigma$, corresponding to the induced flow $r \circ \varphi$.

4.3. **Local stability.** Recall from Section 2 that a game $g$ is regular if $h$ is differentiable at each point in $h^{-1}(g)$. Also, for a regular game $g$, the degree of an equilibrium $\sigma$ is the sign of the determinant of $Dh$ at $z = H(g, \sigma)$, or equivalently of $D\psi$ since $\psi(z) = h(z) - g^*$. Assume now that the given game $g^*$ is regular, and consider an equilibrium $\sigma^*$ of $g^*$. Let $z^* = H(g^*, \sigma^*)$. Choose a neighbourhood $V$ of $z^*$ such that the sign of the determinant $\text{Det}(D\psi)$ is constant on $V$. By Theorem 4.3, for generic $z_0 \in V$ the trajectory (in forward time) starting at $z_0$ converges to a zero in $Z^*$ that $r$ maps to an equilibrium of $g^*$. If $\text{Deg}(\sigma^*) = +1$ then the induced trajectory in $\Sigma$ converges to it. If $\text{Deg}(\sigma^*) = -1$ then the induced trajectory converges to another equilibrium of $g^*$, since $g^*$ is generic. Thus we have shown

**Theorem 4.4.** If $g^*$ is generic then for each equilibrium $\sigma^*$ of $g^*$ there exists a neighbourhood $V$ of $H(g^*, \sigma^*)$ such that the induced trajectory converges to an equilibrium of $g^*$ for almost every initial condition in $V$.

That is, for each generic $z_0 \in V$, starting from $z_0$ the induced trajectory either

1. Converges to $\sigma^*$, and this happens if and only if $\text{Deg}(\sigma^*) = +1$, or
2. Converges to a different equilibrium $\sigma' \neq \sigma^*$.

This generalizes Shapley’s (1974) demonstration for the Lemke-Howson algorithm applied to two-player generic games that $(+1)$-equilibria are ‘sinks’ and $(-1)$-equilibria are ‘sources’. Running time backward gives the opposite result, which has practical implications for computation. If one finds first a $(+1)$-equilibrium and then restarts from an initial condition nearby with the system run backwards (with time reversed) then one finds a $(-1)$-equilibrium of the same game if the trajectory does not diverge. This tactic is familiar in applications of the Lemke-Howson algorithm: by alternating rays and time directions one can find many equilibria, but not necessarily all.
4.4. Computing the orientation $\theta_T$. In this subsection we give a practical formula for determining the orientation $\theta_T$. Consider the vector field

$$\dot{z} = -[\text{Adj} \ D\psi_z] \cdot \psi(z),$$

where Adj indicates the adjoint matrix of the Jacobian $D\psi_z$, i.e., the transpose of the matrix of cofactors. If $z$ is a regular point of $\psi$ then $\dot{z}$ and $\theta_T$ point in the same direction; hence, $\theta_T = \dot{z}/\|\dot{z}\|$. More generally, they point in the same direction if $z$ is a regular point of $f$ such that $[\text{Adj} \ D\psi_z] \cdot \psi(z)$ is nonzero. Since the set of points where $[\text{Adj} \ D\psi_z] \cdot \psi(z)$ is zero is a set of codimension 2, one can use the vector field (1) for computations. In particular, if $h(z^*) = g^*$ then $\psi(z^*) = 0$, so $z^*$ is a zero in $Z^*$ and a rest point of the vector field (1); equivalently, $z^* = H(g^*, \sigma^*) \in Z^*$, where $\sigma^* = r(z^*)$ is an equilibrium of the game $g^*$.

The vector field (1) provides a dynamical system for Smale’s GNM. In Section 5 we use a variant adapted to efficient computation. As emphasized by Keenan (1981), the system (1) is preferable to the usual form of the Newton method in which $\dot{z} = -(D\psi_z)^{-1} \cdot \psi(z)$ because it does not rely on the existence of the inverse of the Jacobian matrix $D\psi_z$; in particular, (1) does not halt necessarily at a singularity of $D\psi$ of codimension 1. Indeed, the adjoint of an $m \times m$ matrix has rank either $m$ or 1 as long as the matrix has rank $m - 1$ or more and is the zero matrix otherwise. Thus, other than rest points, the system (1) halts only where it encounters a singularity of codimension $\geq 2$. This advantage is crucial because singular points of codimension 1 can occur along every generic ray from the true game, but singularities of codimension $\geq 2$ are nongeneric.

The fact that at each point on the trajectory the orientation is determined from the sign of the determinant of the Jacobian, and this sign is locally constant, has useful implications for computation. The Lemke-Howson algorithm and its extensions to many-player games by Rosenmüller (1971) and Wilson (1971) take advantage of this fact implicitly by following a specified direction along each segment of the path until a sign reversal is signaled by arrival at a point in the strategy space where continuation would make an unused strategy optimal or a used strategy’s probability negative, in which case the continuation’s direction on the next segment is indicated by the requirement that the newly optimal strategy’s probability increases or the newly suboptimal strategy’s payoff decreases relative to the optimal payoff. The same technique applies in our setup because at each juncture the retraction $r$ traces the associated mixed strategy $\sigma = r(z)$.

5. Implementation of GNM

This section describes pseudocode for the version of GNM in Section 4.4. Our implementation finds all equilibria accessible by extending the linear homotopy from the ray to the entire line through the true game. Using this extension, the algorithm calculates an odd number of equilibria with alternating indices +1 and −1 until it has found all equilibria accessible via the prescribed line. Thus in the situation depicted in Figure 2 it finds all three equilibria, regardless of whether it begins at the far left or the far right, because it traverses the entire S-shaped graph before exiting along the ray opposite to the one on which it entered. A complete implementation is fully described and listed in the technical Appendix posted on the journal’s website at [www.nyu.edu/jet/supplementary.html](http://www.nyu.edu/jet/supplementary.html).

The algorithm implements the dynamical system (1) specified in Section 4.4 — but with one minor change motivated by extensive computational testing to find a version that is numerically stable and converges quickly. We use the orientation and flow implied by the vector field

$$\dot{z} = -u(z)[\text{Adj} \ D\psi_z] \cdot \psi(z),$$

where $u(z)$ is a function designed to improve numerical stability and convergence.
where \( u : \mathbb{R}^n \setminus Z^* \to \mathbb{R} \setminus \{0\} \) is a map indicating the local velocity. Adding a velocity parameter leaves unchanged the associated orientation \( \theta_T(z) = \dot{z}/\|\dot{z}\| \) at each \( z \in f^{-1}(b) \) for which \( T \) is the support of the associated strategy \( \sigma = r(z) \).

The algorithm implements (2) as follows. Along the path, at each time \( t \) the current state \( (z(t), \lambda(t)) \) is a zero of the equation \( \psi(z(t)) - \lambda(t)b = 0 \), or equivalently \( \sigma(t) \equiv r(z(t)) \) is an equilibrium of the game \((\tilde{G}^*, g^* + \lambda(t)b)\). Therefore, \([D\psi_z] \dot{z}(t) - \dot{\lambda}(t)b = 0\), where one allows different directional derivatives on the two sides of a boundary where \( z(t) \in \partial X(T) \cap \partial X(T') \). This equation is consistent with (2) if \( \dot{\lambda}(t) = -\text{Det}(D\psi_z) \) and the velocity is \( u(z(t)) = 1/\lambda(t) \). Therefore, the algorithm implements the dynamical system

\[
\dot{z}(t) = -[\text{Adj} D\psi_z] \cdot b \quad \text{and} \quad \dot{\lambda}(t) = -\text{Det}(D\psi_z),
\]

starting from an initial state \( (z(0), \lambda(0)) \) at \( t = 0 \) for which \( \lambda(0) \) is sufficiently large that \( \sigma(0) = r(z(0)) \) is the unique (pure-strategy) equilibrium of the game \((G, \lambda(0)b)\). Note that the velocity \( u(z(t)) \) is negative where \( \lambda(t) \) is negative, which has the convenient implication that there is no need to run time backwards over the half-line where \( \lambda \) is negative. The implication that the speed \( |u(z(t))| \) increases as the trajectory approaches a zero \( z^* \in Z^* \) is necessary to ensure quick convergence when using discrete computations.

For computational purposes, time is divided into discrete increments of duration \( dt \), so the corresponding increments are

\[
dz(t) = -dt \times [\text{Adj} D\psi_z] \cdot b \quad \text{and} \quad d\lambda(t) = -dt \times \text{Det}(D\psi_z).
\]

Implementation of this dynamical system is entirely straightforward. Among versions we tested, even the simplest works well and this is the one described in subsection 5.3 and for which numerical results are reported in subsection 5.4. Accuracy is sustained if provision is made to correct for accumulated errors periodically (see the three options in subsection 5.2, and Steps 9 and 11 of the pseudocode in subsection 5.3). A minor complication is that at a boundary point the Jacobian \( Dr \) of the retraction changes discontinuously and therefore so does the Jacobian \( D\psi \). Therefore, along the path through the domain \( X(T) \) for the current support \( T \) of \( r(z(t)) \), one needs to estimate the next time \( \hat{t} \) when \( z(\hat{t}) \in \partial X(T) \) where the support \( T \) changes. Our numerical results are based on the simplest linear extrapolation (see Step 6 of the pseudocode), which works well when the time increment \( dt \) is sufficiently small. A nonlinear extrapolation allows a larger step \( dt \) and thereby accelerates the algorithm, but this extension is omitted here.

### 5.1. Input data.\

The point \( b \in S \) that specifies the ray of the homotopy is assumed to be generic so that there is no branch point on the resulting trajectory, except possibly at the true game if it is nongeneric. The true game is assumed to be located at the origin, \( g^* = 0 \), so the game is specified simply by its payoff function \( G^* \) rather than the pair \((G^*, g^*)\) used for Theorem 2.1 — and notation is simplified further by using \( G \) rather than \( G^* \). The input data are:

- A small stepsize \( dt > 0 \) for the time parameter. Our implementation interprets \( dt \) as a fraction of the interval required to reach the next boundary, as estimated from a linear extrapolation.

- A generic vector \( b \in S \setminus C \) that defines the ray for the path of the homotopy. The ray is interpreted as originating from the true game located at the origin of \( \mathbb{R}^m \).

- A subroutine \textit{Payoff} that computes for each mixed-strategy profile \( \sigma \in \Sigma \) the Jacobian matrix \( DG(\sigma) \) of the vector \( G(\sigma) \in \mathbb{R}^m \) of expected payoffs from the players' pure strategies.
Because \( G(\sigma) \) is homogenous of degree \(|N| - 1\) (ignoring the restriction \( \sigma \in \Sigma \)), Euler’s Theorem implies that \( G(\sigma) = DG(\sigma) \cdot \sigma / (|N| - 1) \) so expected payoffs can be computed from the Jacobian \( DG \) via this inner product. Also, the Jacobian of \( \psi \) is calculated as \( D\psi(z) = I - (I + DG(r(z))) \cdot Dr(z) \), where \( I \) is the identity matrix and \( Dr(z) \) is locally constant, depending only on the support of \( r(z) \). The Jacobian \( DG \) has a simple interpretation: the element of \( DG \) in player \( n \)'s row \( s \in S_n \) and player \( \hat{n} \)'s column \( \hat{s} \in S_{\hat{n}} \), where \( n \neq \hat{n} \), is the expected payoff to player \( n \) from his pure strategy \( s \) when player \( \hat{n} \) uses his pure strategy \( \hat{s} \) and all other players use their mixed strategies in \( \sigma \).

If the game is in normal form then the payoffs from pure-strategy profiles can be stored as an \((|N| + 1)\)-dimensional array in which a typical element \( G_n(s_1, s_2, \ldots, s_{|N|}) \) is the payoff to player \( n \in N \) from the pure-strategy profile \((s_1, s_2, \ldots, s_{|N|})\). In this case, the calculation of \( DG \) comprises a series of inner products; see the formula for \( v_s(\sigma) \) in Section 2. Numerical stability can be enhanced by re-scaling the payoffs, but it is sufficient to apply re-scaling only to the Jacobian \( DG \). From the normal form it is easy to identify quickly all pure-strategy equilibria, so our implementation routinely does that first. If the game is given in extensive form then the subroutine \textit{Payoff} can take advantage of the structure of the game tree and the smaller dimension of the space of extensive-form payoffs. Alternatively, the game can be formulated in agent-normal form.

The storage requirements and computational times for normal-form games are onerous. For instance, a 5-player game in which each player has 8 pure strategies (i.e., \( m = 40 \)) requires specification of \( 5 \times 8^5 = 163840 \) payoffs, and a naive calculation of \( G(\sigma) \) requires 655360 multiplications. The 2.7 hours required to solve one such problem on a Pentium III computer was absorbed mostly by the \textit{Payoff} subroutine. (Our implementation in the Appendix is programmed in the interpretative language APL, which is not compiled like C and Fortran, so it usually takes 3 to 10 times longer than a comparable program written in C.) We are not discouraged by these facts because we suppose that interesting applications will be specified in extensive form, and therefore allow faster execution of \textit{Payoff}.

Our implementation allows calculation of all equilibria accessible via the ray specified by \( b \). Consequently, if additional equilibria are desired then an additional input parameter is the minimum value \( \lambda_\ast \) of \( \lambda(t) \) that signals termination if the last previous equilibrium had a positive index. If \( \lambda_\ast \ll 0 \) then the program continues finding equilibria with alternating indices +1 and -1 until \( \lambda(t) < \lambda_\ast \) and the previous index was +1. If the true game \( G \) is nongeneric, perhaps because it derives from an extensive form, then the index might actually be 0 and/or the equilibrium found might be only one point in a connected component of equilibria, but presently our implementation provides no definitive signal that this is the case. The number of equilibria found is often quite large (dozens for large games), as suggested by von Stengel’s (1997) results for the 2-player case.

### 5.2. Useful subroutines

Calculations conveniently isolated in two additional subroutines are:

- **Retract**: Calculates the retraction \( \sigma = r(z) \) and its Jacobian \( Dr \equiv R \in \mathbb{R}^{m \times m} \), which depends only on the support \( T \) of \( \sigma \). Represent the support, or basis, by a Boolean vector \( B(z) \) with \( B_s(z) = 1 \) iff \( \sigma_s > 0 \). Note that the support changes at each boundary; that is, some \( B_s(z) \) switches from 0 to 1 or from 1 to 0. Execution of \textit{Retract} is very fast.

- **Adjoint**: Calculates the adjoint and determinant of a square matrix. This is done by Gaussian elimination, which can be time-consuming for large matrices, but recall from above that ordinarily it is the function
Payoff that is the most time-consuming. Gaussian elimination requires approximately $m^3$ multiplications and divisions, which can be small compared to the number $|N| - 1|N|m^2|N|$ required by Payoff when $|N| > 3$ and the game is in normal form.

An important feature of our present implementation is absence of procedures to resolve degeneracies (branch points on the path), on the assumption that the ray is specified by a generic point $b \in S \setminus C$. Recall from Proposition 3.1 that genericity of the ray is sufficient for absence of branch points even if the true game $G$ is nongeneric. However, our present implementation makes no attempt to identify branch points that would signal that the chosen point $b$ is not generic.

To speed calculations and improve numerical precision, our implementation allows the user to invoke any subset of the following three optional modifications, even though any of these might invalidate application of Proposition 3.1.

- **LNM**: Because we want our implementation to allow a large stepsize, we face the practical problem of correcting accumulated errors due to the large stepsize, as well as the usual problems with accumulated roundoff errors. Therefore, the implementation allows a user the option to specify that periodically a subroutine that implements the local Newton method (LNM) is invoked to regain a position on the graph above the ray at $\lambda(t)b$. That is, starting from $z(t)$, LNM solves the equation $z = r(z) + G(r(z)) + \lambda(t)b$ for $z$ to replace $z(t)$ in the continuation. This is done using the Jacobian $DG$ at $z(t)$, so the procedure is actually a quasi-Newton method, and of course it cannot be applied where the Jacobian is nearly singular, as often happens at a boundary. Invoking LNM only periodically runs the risk of invalidating Proposition 3.1 because the trajectory could drift away from the intended path sufficiently to enter a cycle (like the one in Figure 2). LNM with $\lambda(t) = 0$ is used as the final step of the algorithm to ensure a highly accurate solution; e.g., our numerical results are based on a requirement that the maximum error is less than $10^{-6}$ in terms of payoffs.

- **Wobbles**: Upon reaching a boundary where the support changes, we allow the user to account for accumulated errors by perturbing the point $b$ so that the current point $(G, \lambda(t)b, r(z(t)))$ is in the equilibrium graph $E$. This allows a large final step to reach the boundary, with the error absorbed by perturbing the ray. LNM cannot be used routinely for this because at a boundary the Jacobian $DG$ is often singular or nearly so (and if $|N| = 2$ then it is always singular at the second, fourth, sixth ... boundaries encountered, but not at the odd ones, as in the Lemke-Howson algorithm that alternates adjustments of the strategies of the two players). Computational experience indicates that such “wobbles” of the ray speed calculations and enable in a simple way the implementation to be robust to accumulated errors; also, the wobbles required are typically quite small. However, in principle it introduces the possibility of cycling, so a subsequent implementation might include an explicit error correction procedure, or a more sophisticated method of extrapolating the path to the next boundary that uses polynomial extrapolation of degree $> 1$.

- **Perturbed Jacobian**: Our present implementation avoids strictly vertical segments (i.e., $z(t)$ moves while $\lambda(t)$ remains fixed because the Jacobian $D\psi$ is singular) by inserting a small $\epsilon \approx 10^{-12}$ into the blocks of zeros on the diagonal of the Jacobian $DG$. This amounts to tilting the trajectory slightly so that no vertical steps are necessary. Its theoretical justification stems from the analog to computing symmetric equilibria of symmetric games in which there is a small payoff from interactions between two players.
assigned the same roles. This simple device works exceedingly well, with nary a problem in our experience. However, a subsequent implementation might be designed explicitly to move vertically so that there is exact conformity with the theoretical construction in Sections 3 and 4. The obvious candidate is to use a full-rank submatrix of $DG$, which is how the Lemke-Howson algorithm does it implicitly for two-player games by alternating between the two submatrices of the players’ payoffs.

5.3. **Pseudocode for the algorithm.** The pseudocode proceeds as follows (with parenthetical remarks about our implementation):

1. Initialize: Identify for each player $n \in N$ the unique pure strategy $s^*_n = \text{Arg max}_{s \in S_n} b_s$ so that the pure-strategy profile $s^* = (s^*_n)_{n \in N}$ is the unique equilibrium when $\lambda$ is sufficiently large, say $\lambda > \lambda^*$. Compute $\lambda^*$ by checking the inequalities that ensure that $s^*$ is an equilibrium. Let $\sigma^* \in \Sigma \subset \mathbb{R}^m$ be the mixed strategy with support $s^*$. Start with $t = 0$, $\lambda(0) > \lambda^*$, and $z(0) = z^* \in \mathbb{R}^m$, where

$$z^* = H(G, \lambda(0)b, \sigma^*) = \sigma^* + G(\sigma^*) + \lambda(0)b,$$

and the initial support is $B^* = \sigma^*$.

2. Call **Retract** to compute the $m$-vector that is the mixed strategy $\sigma = r(z(t))$, and to identify the support $B = B(z(t))$ of $\sigma$ at $z(t)$. Using $B$, compute the $m \times m$ matrix $R$ that is the Jacobian $Dr$ of the retraction $r$ at $z(t)$.

3. Call **Payoff** to compute the $m \times m$ matrix $DG$ that is the payoff’s Jacobian at the mixed strategy $\sigma$. Call **Adjoint** to compute the $m \times m$ adjoint matrix $J = \text{Adj}(D\psi)$ and scalar determinant $\text{Det}(D\psi)$ of $\psi$’s Jacobian $D\psi = I - (I + DG) \cdot R$ at $z(t)$, where $I$ is the $m \times m$ identity matrix.

4. Set $\dot{z}(t) = -J \cdot b$ and $\dot{\lambda}(t) = -\text{Det}(D\psi)$. Recall that these are the key equations of the implementation of the global Newton method via the dynamical system (3) above.

5. Compute the $m$-vector $v(t) = DG \cdot \sigma / (|N| - 1) + \lambda(t)b$ of the pure strategies’ expected payoffs, and its time-derivative $\dot{v}(t) = DG \cdot R \cdot \dot{z}(t) + \dot{\lambda}(t)b$, but restrict these to those pure strategies in the support; i.e., use only the players’ optimal expected payoffs.

6. Using $(v(t), z(t))$, and supposing $(\dot{v}, \dot{z})$ were constant, extrapolate to the next boundary where some pure strategy $\hat{s}$ will enter or leave the support. That is, calculate the maximum time interval $\Delta > 0$ for which $z(t) + \Delta \dot{z} \geq v(t) + \Delta \dot{v}$. (It is expedient in the initial step away from a boundary to exclude from this inequality the row of the pure strategy, say $s$, that entered or left the support in the previous iteration, since even though $\sigma_s = 0$ identically there [by definition of the boundary], its payoff $v_s$ and therefore $z_s$ may have been calculated with an error sufficient to imply that the constraint for $s$ determines $\Delta$. We do this on the presumption that it is unlikely [but perhaps not impossible when $|N| > 2$] that $s$ defines also the next boundary encountered on the path.) If $\Delta = \infty$, as can happen when $\lambda, \ll 0$ and at $\lambda(t) < 0$ there are no further equilibria on the path, then exit with an indication that all equilibria on the path have been found.

7. If $|N| = 2$ then the constancy supposition in Step 6 is correct so use the full stepsize $\Delta$. If a criterion for proximity to the next boundary is met, then also use the full stepsize $\Delta$ so that a jump to the estimated boundary is made in the next step. (Our present implementation uses a crude criterion for proximity: it jumps to the next boundary after a fixed number $k$ of steps, where $dt > (1 - dt)^k$.)
either of these cases, set $\delta = \Delta$. Otherwise, the stepsize is $\delta = dt \times \Delta$, where the initial datum $dt$ is interpreted as a fraction of the estimated interval $\Delta$ to the next boundary.

8. If $\lambda(t) + \delta\dot{\lambda}(t) \leq 0$ (or when $\lambda_* \ll 0$), reverse this inequality if the previous equilibrium found had index $+1$ so the next must have index $-1$), then infer that an equilibrium exists at $\lambda(t + \delta) = 0$ where $\delta = -\lambda(t)/\dot{\lambda}(t)$, so go to Step 11 to record this fact. Otherwise, move to $z(t') = z(t) + \delta \dot{z}(t)$ and $\lambda(t') = \lambda(t) + \delta\dot{\lambda}(t)$ at $t' = t + \delta$. If $\lambda(t') < \lambda_* \ll 0$ then exit with an indication that the search for additional equilibria has terminated.

9. Reset $t$ to the value of $t'$. If $\dot{\delta} = \Delta$ (the first two cases in Step 7), then a boundary has been reached so go to Step 10. Optionally, if a specified number of iterations have passed then correct for accumulated errors by calling $LNM$ to re-establish a position above the ray; that is, solve the equation $z - r(z) - G(r(z)) - \lambda(t)b = 0$ for $z$, which then replaces $z(t)$. Optionally, return to Step 5 without computing new values for the Jacobians (as in a quasi-Newton method), but note that this option is useless unless $v(t)$ and therefore $G((r(z))$ is computed explicitly without relying on the previous Jacobian $DG$. (To clarify their interpretation, none of the numerical results we report use either of these options.) Otherwise, return to Step 3 where $DG$ is computed anew before the next step.

10. Having reached a boundary, record the change in the support $B$; that is, the pure strategy $\hat{s}$ found in Step 6 either enters or leaves the support.

11. Optionally, to ensure that $z(t)$ corresponds to an equilibrium of the game $(G, \lambda(t)b)$, perturb the ray to $b = (z(t) - r(z(t)) - G(r(z(t)))/\lambda(t))$. Return to Step 2.

12. Having found an approximate equilibrium, call $LNM$ to improve the numerical precision. Record the equilibrium that has been found, and its index. Return to Step 2 to continue the path to the next equilibrium with opposite index.

This pseudocode corresponds closely to the implementation listed in the Appendix. Our numerical results did not invoke either option in Step 9. Most of the numerical results invoked the wobbling option in Step 11 when $N > 2$. The actual code that was used differed immaterially from the one in the Appendix, which has been edited to conform to the pseudocode listed above. The linear extrapolation in Step 6 and the proximity criterion in Step 7 are effective but crude – one can hope for superior results from refinements.

5.4. **Computational experience.** The following table shows the average time in seconds (to four digits accuracy) required on a Pentium III computer for our program to reach the first equilibrium on the path to a normal-form game with $|N|$ players, each of whom has the same number $m_n$ of pure strategies. After the first equilibrium is found, the others accessible via the ray are usually found quickly — but not of course when there are very many (some games in our sample had more than 30 equilibria accessible via the chosen ray). The lower-right portion of the table is empty because normal-form games of that size exceed the active memory of the computer used. For each case but one, twenty examples were randomly generated, as were the rays, so presumably each is generic; only one example was run with $|N| = 5$ and $m_n = 8$. Many other examples were solved, but the times shown are for the last twenty examples solved. In every case the stepsize parameter was $dt = 0.02$ and the solution was accurate within $10^{-6}$ in terms of the payoffs. The evident exponential growth in solution times reflects mainly the time required to compute the Jacobian $DG$ when the game is specified in normal form. Two-player games are solved quickly because the algorithm jumps...
linearly from one boundary to another, following exactly the homeomorphic image (under $H$) of the path of the Lemke-Howson algorithm.

| $|N|$ | $m_n$ : | 2   | 4   | 6   | 8   | 10  | 12  | 14  | 16  |
|------|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| 2    | 0.0070  | 0.0140 | 0.0370 | 0.0791 | 0.2373 | 0.3281 | 0.5358 | 1.2820 |
| 3    | 2.2530  | 8.9950 | 33.0900 | 143.1000 | 251.2000 | 536.6000 | 853.3000 | 1471.0000 |
| 4    | 3.6500  | 25.8200 | 144.5000 | 469.2000 | 854.9000 |
| 5    | 10.4200 | 126.3000 | 1116.0000 | 9745.0000 |
| 6    | 19.6700 | 315.9000 |

These examples include games much larger than those reported by Herings and Peeters (2001) for their Fortran implementation of a differentiable homotopy for the linear tracing procedure of Harsanyi and Selten. Our times for those games of the same size are faster by a factor of about 3 for $|N| > 3$. GNM has the further advantage that it can compute all equilibria accessible via each specified ray. In a companion paper (Govindan and Wilson, 2001a) we present another algorithm based on polymatrix approximation of the payoff function. This algorithm is useful as a fast start for GNM because it reaches quickly the vicinity of the first equilibrium on the trajectory. It shortens times by factors of 2 or more (10 on large problems), enables solutions of games with as many as 12 players, and overall exhibits somewhat slower exponential growth.

The present implementation requires the game to be in normal form. A computationally efficient implementation of the GNM algorithm for extensive-form games involves additional issues. The goal of first importance is an economical representation of the game tree and payoffs to reduce storage requirements and to speed computation of the Jacobian $DG$, following the direction developed in von Stengel (1996) for 2-player games. A further task is to devise a method of identifying components of equilibria, since these are prominent features of such games, and computing their indices, since a nonzero index is a sufficient condition for stability in the sense of Mertens (1989). For very large games, the method in Wilson (1972) might be adapted to improve computational efficiency by following the GNM path for only a subset of the pure strategies until one outside this subset becomes an optimal reply to those inside.

References


Figure 1: Schematic representation of the maps and points in their domains and ranges.
Figure 2: Schematic representation of a section of the equilibrium graph above a line through the game positioned at the origin.