Computation in Finite Games: Using Gambit for Quantitative Analysis

BY THEODORE L. TUROCY

Department of Economics
Texas A&M University
College Station TX 77843

EMAIL: turocy@econmail.tamu.edu

April 20, 2005

1 Introduction

Richard McKelvey, Gambit, and computing in finite games

Richard McKelvey, in applying noncooperative game theory and quantitative analysis to problems in political science, was naturally confronted with the need for computer programs to assist in the analysis and solution of games. During the 1980s, Richard began to write computer programs to compute Nash equilibria of games, initially in the language BASIC and later in C. These programs were the ancestors of the library of routines for analyzing finite games that came to be known as Gambit.[1]

Gambit owes its current form to a redevelopment and expansion during 1994 and 1995 under a National Science Foundation grant awarded to Richard jointly with Andrew McLennan, with this author serving as principal programmer. The project currently comprises an extensive library in C++ to represent and analyze finite extensive and normal form games, as well as a graphical user interface to manipulate games and visualize results. Gambit is continuing to be developed as open source software, and is distributed freely under the terms of the GNU General Public License.

This chapter explores some of Richard's contributions to the quantitative analysis of finite games. In addition, it provides for the researcher interested in pursuing a research agenda similar in style to Richard's a quick introduction to the existing methods for computing Nash equilibria in finite games. Relative to the excellent surveys of MCKELVEY AND MCLENNAN [19] and VON STENGEL [32], the goal is to provide some practical guidance, based on the author's experience in the development and maintenance of Gambit, in the selection of the best methods for computing equilibria, and formation of reasonable expectations as to how large and what types of games are feasible to analyze numerically.

1. Richard chose the name "Gambit" as a play on the words "game" and "bit," the fundamental unit of information.

Why compute?

Regardless of one's ultimate objectives, a handy use of computation in analyzing a formal model is the ability to build intuition about the model's behavior as its structure and parameters are changed. This process allows the modeler to play with the model, and begin formulating hypotheses both about the overall predictions and implications of the model, as well as for possible general results in the model. The intuition built from these experiments may lead to insights, which can subsequently be develped into a formal expression as a theorem. Even when a simple characterization in the form of a theorem turns out not to be available, computational experiments yield information about the quantitative predictions of the model.

The computation of Nash equilibria in finite games is a tedious process, as anyone who has manually computed equilibria in a game with more than two players or involving more than two strategies per player will attest. In fact, the number of equilibria can grow quickly in the size of the game (see, for example, McKelvey and McLennan [20]). The precise identification of the formal computational complexity of computing Nash equilibria in finite games remains today an active area of research. What is clear, however, is that the problem is computationally difficult, meaning that automating the process is essential.

One application of formal game models involves the determination of unknown parameters based on observed data. This process requires knowledge of the equilibria of many closely-related games. For example, consider the McKelvey and Palfrey [21] experimental study of behavior in the centipede game. This model introduced a probability that a player has "altruistic" preferences. This rendered the centipede game an extensive form game of imperfect information, with a parameter, the probability of these preferences occurring, to be estimated from the data. What is observed in those data is the frequency with which a player chose to "pass;" from that, one backs out the underlying probability. However, the mapping from game parameters to equilibrium properties need not be straightforward, or even have a closed form; hence, iteratively computing the equilibria of the game for different probabilities is necessary to implement the estimation procedure. This idea is generarlized by, for example, the recent work of Bajari et al [2], who incorporate methods for computing all Nash equilibria of a game in an estimation procedure to determine underlying parameters of a game.

An alternative identification approach uses the quantal response equilibrium concept of McKelvey and Palfrey [22], [23]. The QRE is a staple of analysis of finite games in laboratory experiments (for example, the survey of Goeree and Holt [7] gives a set of examples where quantal reponse equilibrium predictions predict qualitative features of laboratory data). Quantal response equilibria can be expressed in closed form only in special cases, thereby making the problem of doing estimation using QREs inherently numerical.

Reasonable expectations

As with any endeavor to numerically analyze a problem, it is best to start simple, attempting to work with "out-of-the-box" procedures available in existing packages. This chapter summarizes some of the procedures which are applicable to any finite game, and which are likely to produce results for most games of small or medium size.

It is important to keep in mind that, on average, the time required to find equilibria increases rapidly in the size of the game. Further, since the increase in running time may be exponential in the size of the game, future marginal advances in computing technology, such as processor speed or multiprocessing, may not have a significant impact on the size of game which may be adequately handled by general-purpose methods. In these cases, practical numerical analysis of a game will require techniques which avail themselves of the specific structure of the game.

Such methods are currently a topic of active development. The simplest example of reducing the effective size of a game for computational purposes is to consider symmetric games and search only for the symmetric equilibria of the game. Other classes of games have structure to their payoffs which can be exploited computationally. ECHENIQUE [5] gives a procedure for efficiently computing all equilibria of supermodular games. KEARNS ET AL [13], among others, consider games with a particular graphical structure identifying the local interactions among players. Similarly, polymatrix games (see GOVINDAN AND WILSON [9]), where players' payoffs are determined only by bilateral interactions, have a computationally easier structure.

2 Numerical Methods to Compute Nash Equilibria

Some general considerations

MCKELVEY AND MCLENNAN [20] observe that the definition of a Nash equilibrium may be recast in a number of different mathematical formulations. Each formulation suggests a different approach to numerically computing Nash equilibria. Since these methods are quite distinct, they are often adapted to different purposes.

A first observation is that if a game has an extensive (game tree) structure, it is generally advisable to choose methods which operate on the extensive game, rather than on its reduced normal form representation. The number of strategies in the reduced normal form of an extensive form game may grow quite (infeasibly) large for extensive forms of modest size; or, to put it another way, there is redundancy in the reduced normal form representation. The worst-case scenario for this growth is the case of an extensive form game which has many "parallel" information sets; that is, a pair of information sets for a player such that play passes through one or the other of them. The extreme examples of games like this are sequential-move games of perfect information such as tic-tac-toe.

Independent of one's ultimate objectives, the first step in analyzing a game numerically is to identify and remove from consideration dominated strategies or actions. GILBOA ET AL [6] show that computing strictly dominated strategies is of a complexity easier than that of computing a Nash equilibrium. Iteratively eliminating strictly dominated strategies, when possible, helps reduce the size of the game the equilibrium computation methods must confront, without removing any Nash equilibria. If one's objective will be only to compute a single equilibrium, iterative removal of weakly dominated strategies is also suggested.[2]

The second step in searching for equilibrium is to perform a search for all equilibria in pure strategies. While there is no procedure for doing this more sophisticated than a brute-force search over all possible combinations of strategies or actions, it is effective even for fairly large games. As an added benefit, equilibria in pure strategies are often considered the most compelling as a prediction of how a game would be played, so identifying them quickly is of practical interest as well.

Methods for all finite games

If a game has no equilibria in pure strategies, or if a fuller characterization of the equilibria of a game involving randomization is desired, more sophisticated approaches are indicated. Methods which are applicable to all games, regardless of the number of players or payoff structure, are considered next. When these methods are implemented in Gambit, their Gambit names are indicated in `typewriter` text.

One method generalizes the idea of looking first for equilibria in pure strategies. Porter et al [25] present a heuristic-based approach (`PNS`) for searching for equilibria which looks first for equilibria in which players give positive probability to as few strategies as possible. This method can be used to find all equilibria, but it is designed to be biased towards finding a first equilibria as quickly as possible; this makes it ideal for searching for either one equilibrium, or two equilibria (to determine whether the game has a unique equilibrium). Since this method first searches for pure-strategy equilibria by design, doing an independent search for pure-strategy equilibria is not necessary when using this approach.

The `PNS` idea of searching over particular supports (i.e., sets of strategies played with positive probability) can be adapted to instead organize supports in a search tree. This method (`EnumPoly`) uses a top-down search that is able to prune some parts of the tree from consideration. Thus, it is better suited in general than `PNS` for the task of computing all equilibria.

In constrast, several methods operate by attempting to compute increasingly better approximations to a Nash equilibrium. For applications where the equilibria of interest involve randomized strategies, approximation methods may be attractive in that the sequence of strategy profiles computed by the algorithm will converge in the limit to an equilibrium. As such, the intermediate output of these methods can be viewed as giving some hint as to where an equilibrium might lie. The support-based methods outlined in the previous paragraphs give no indication whether the next support to be considered is more or less likely to contain an equilibrium.

One robust approximation method is the simplicial subdivision method for normal form games by van der Laan et al [30] (`Simpdiv`). This procedure constructs a grid over the space of mixed strategy profiles, similar to the algorithm of Scarf [27]. This grid is then traversed to compute a fixed point on this grid. Once such a fixed point is found, the grid may be refined, and the fixed point of the coarser grid used as a starting point for a search on the refined grid. In the limit as the spacing between grid points decreases to zero, this process should compute a Nash equilibrium.

2. These comments assume the ultimate goal is identification of Nash equilibria. Some behavioral solution concepts, quantal response equilibria among them, are affected by the presence of dominated strategies.

Each fixed point computed as the grid is refined can be viewed as computing a sequence of approximate equilibria, which in the limit will tend to a Nash equilibrium. In practice this is often the case, though there is no theoretical guarantee that the process will converge quickly. It is possible to appear to be converging to a particular point, only to have the computed approximate equilibrium change substantially as the grid is refined. Further, it is possible that a profile that appears to be the limit of a sequence of approximate equilibria is only revealed not to be the limit at a very fine grid size.

At every grid size, it is guaranteed that a fixed point on that grid will be found; however, on some games a very long path may be traversed to locate it. In the implementation of simplicial subdivision in Gambit, an optional "leash" parameter restricts how far afield the algorithm searches, on the principle that it is possible but unusual for an approximate fixed point to "disappear" as the grid is refined. Use of this restriction sometimes speeds convergence of the algorithm; however, it also means that the algorithm with this leash activated is not guaranteed to compute an approximate equilibrium.

The starting point for the simplicial subdivision can be any point on the initial grid. The resulting equilbrium found depends on the choice of the initial condition; therefore, it is possible to find distinct equilibria using this method by choosing different starting points. However, since the relationship between the initial point and the computed equilibrium is not easy to see, it is not possible to be sure one has found all equilibria using this approach.

Another method that computes a progressively better approximation to an equilibrium is inspired by the quantal response equilibrium concept of McKelvey and Palfrey [22], [23] (`Logit`). Each branch of the logit quantal response equilibrium correspondence converges in the limit to a Nash equilibrium of the game as the randomness in payoffs is decreased to zero. A method to trace a branch of the correspondence efficiently is proposed in Turocy [29], which is the basis for the implementation in Gambit. Asymptotically as the noise is decreased, the quantal response equilibria converge fairly rapidly to the limiting Nash equilibrium. As the quantal response equilibrium concept is now widely used for quantitative analysis of data from laboratory experiments, computation of Nash equilibria by this method has the advantage that the points necessary for such estimation are computed as a byproduct.

The natural starting point for this method is the profile where all strategies (or actions, for the extensive form version) are played with equal probability, and the noise parameter is infinity; this point is always in the quantal response correspondence for all games. The equilibrium found by traversing from this point is generically unique, and is called the "logit solution" by McKelvey and Palfrey. Computing other Nash equilibria is more problematic. Other, disconnected branches may exist, each of which connects two Nash equilibria (again, generically); it is not clear how to reliably locate such branches in general. In addition, not all equilibria are limits of logit quantal response equilibria, meaning that the method is not guaranteed to compute all equilibria.

Another promising approach to computing equilibria is the global Newton method of Govindan and Wilson [8]. This method is based on facts about how the set of Nash equilibria of a game changes as payoffs are perturbed. It is observed that by making a perturbation large enough, the perturbed game will have a unique equilibrium that is easy to compute. This equilbrium is then traced back as the perturbation is made smaller to reach equilibria of the original game. Since this method may take a long time to reach the vicinity of the original game, Govindan and Wilson [9] introduce a method to reduce this time by approximating the original game by polymatrix games, which are games where the payoffs are determined by bilateral interactions among the players. Games with this structure can be solved using an algorithm due to Lemke and Howson.[3] The global Newton (`Gnm`) and polymatrix (`Ipa`) approaches, while recent developments, currently appear to be promising as the "best" algorithms for computing equilibria for many games.[4]

Another method for approximating a Nash equilibrium is the Lyapunov function method (`Liap`) proposed by McKelvey [18]. This function is based on the differences between the payoff earned by each player at a given strategy profile and the payoff each player would earn by playing his best reply to the other players' choices; the function is therefore nonnegative, and zero exactly at Nash equilibria. Thus, the function is attractive since, approximately speaking, lower values of the Lyapunov function correspond to strategy profiles where players are making best-reply errors that are less costly in payoff terms.

Since the function is differentiable, standard function minimization methods can be used to compute local minima of this function; and, if the function value at the local minimum is zero, then it is a Nash equilibrium. However, local constrained minima of the Lyapunov function exist, and experience indicates that for many games, standard gradient-descent methods will tend to converge to constrained local minima that are not Nash equilibria. The properties of this function are not fully explored, and it may be possible that techniques such as simulated annealing or genetic algorithms, which are designed to operate on functions that may have many local minima, may be effective in establishing Lyapunov function methods as effective means for computing equilibria.

Methods for games with two players

Games with two players enjoy some convenient properties that often make the problem of computing Nash equilibria simpler and more efficient. These derive in large part from the observation that a player's payoff for playing one of his strategies can be expressed as a linear function of the probabilities the other player assigns to her strategies.

The workhorse for computing equilibria in normal form games with two players is the method of Lemke and Howson [16] (`Lcp`). The Lemke-Howson algorithm is a constructive proof of the famous existence theorem of Nash [24]. The method proceeds by following a path of mixed strategy profiles that are "almost" equilibria. Each such profile has the property that exactly one strategy that gives less than the optimal payoff for a player is used with positive probability. At the end of any such path is a profile which satisfies all the conditions of equilibrium. Because of the linearity of each player's payoff function, the Lemke-Howson algorithm can be implemented using matrix "pivoting" procedures, which are well-studied numerical methods.

3. See the next section for details on this algorithm.

4. The Gambit implementation derives from the Gametracer implementation of Blum et al [3].

The general experience is that the Lemke-Howson method finds an equilibrium quickly. However, SAVANI AND VON STENGEL [26] give an example to show that the path that the algorithm takes may be quite long in certain cases. Also, SHAPLEY [28] shows that there exist equilibria that cannot be located using the method; therefore, one can not use the Lemke-Howson method to compute all equilibria.

The specific operation of the Lemke-Howson method is generally opaque, and does not give an easy intuition when compared to other methods. SHAPLEY [28] presents a graphical interpretation of how the method operates on a simple example, though extending that visualization to a larger game may be difficult. Note that while each step of the Lemke-Howson algorithm gives an "almost" equilibrium, in that only one suboptimal strategy is used, it is not in general true that each step provides a successively better approximation to an equilibrium in the sense discussed in the previous section.

The existence of equilibria inaccessible by Lemke-Howson is addressed by an enumeration method (`EnumMixed`) given by MANGASARIAN [17]. This method essentially computes and visits all the strategy profiles that might be visited by the Lemke-Howson method, and therefore can find even those strategy profiles that are not accessible via Lemke-Howson. Therefore, it is suitable in principle for computing all equilibria. However, being an enumeration method, the number of such profiles grows rapidly in the size of the game. The enumeration of these extreme points can be done using the LRS algorithm of AVIS AND FUKUDA [1].

When a two-player normal form game is furthermore constant-sum, it is possible to formulate the conditions for a Nash equilibrium as a linear programming problem (`Lp`). This observation by DANTZIG [4] serves as a constructive proof of the famous Minimax Theorem of VON NEUMANN [31]. Linear programming problems are extensively studied numerical procedures, and many good implementations exist for solving them.

The extensive form does not lend itself as directly to solution via linear programming or linear complementarity programming. An important development in this area is an alternate representation of an extensive form, the sequence form of KOLLER, MEGIDDO, AND VON STENGEL [14]. The sequence form allows formulation of the equilibrium problem in a way that parallels the formulation for normal form games. When the extensive form is zero-sum, characterization of equilibrium points may be done using linear programming in the sequence form. For non-constant-sum games, a variation on Lemke-Howson described by LEMKE [15] may be applied.

Compared to the reduced normal form, the sequence form grows only at the rate the extensive form grows. This occurs because the key concept in the sequence form is a sequence of choices. This representational parsimony combined with the development of efficient implementations of linear programming and linear complementarity programming solvers means that games with more than a million nodes may feasibly be solved, even when much smaller games would be infeasible using normal form methods.

Refinements of Nash equilibrium

Since there may be many Nash equilibria of a game, many "refinement" concepts for Nash equilibria have been proposed to help eliminate Nash equilibria which are deemed to be less plausible.

Several of the algorithms described already are guaranteed to compute Nash equilibria satisfying certain refinements. The implementation of the Lemke-Howson method in Gambit, by virtue of the way degeneracies are handled, is guaranteed to compute equilibria that are trembling-hand perfect. A further refinement of perfection, a proper equilibrium, can be computed using the method of YAMAMOTO [34]. An equilibrium selection method due to HARSANYI AND SELTEN [10] can be implemented using a homotopy method of HERINGS AND PEETERS [11]. An algorithm to compute simply stable sets of equilibria was proposed by WILSON [33].[5]

For extensive games, computation of equilibria satisfying the most basic refinement, subgame perfection, can be accomplished with any algorithm simply by solving each subgame in turn, working backwards from the end of the game. For the additional refinement of sequential equilibrium, the construction of the agent logit quantal response equilibrium and Lyapunov function method for extensive games guarantees the computation of a sequential equilibrium.[6]

3 Looking forward

In their study of the centipede game, published in 1992, McKelvey and Palfrey acknowledged the use of donated time on a Cray XMP supercomputer to accomplish the estimation. Today, similar computations can be carried out on off-the-shelf personal computer hardware. These developments in computing power made the quantitative portion of Richard McKelvey's research program feasible. Indeed, the origins of Gambit lie in Richard's research, and Gambit represents part of Richard's legacy to future researchers seeking to pursue programs patterned on his approaches and methods.

The ubiquity of computers today continues to expand the boundaries of problems that can be tackled with computational tools. JUDD [12] makes a strong case for a significant role of computation in economics, and his comments apply with equal force to the application of rigorous and quantitative methods to political science. As exemplified by much of Richard's work, formal theorem-proving and computation are natural complements. Computational analysis can given initial insights into a model's behavior, which in turn may lead to formal statements of a model's predictions in terms of theorems. Subsequently, computation can play a significant role in the next step, of taking the model to the data in a process of identification of a model's parameters and testing the model's quantitative predictions.

Acknowledgments

This article has been prepared for inclusion in *A Positive Change: The Legacy of Richard D. McKelvey's Most Influential Writings*, Alt, Aldrich, and Lupia, editors. The Gambit website is `http://econweb.tamu.edu/gambit`

5. An implementation of Wilson's algorithm is available from the Gambit website.

6. Note that despite the similar nomenclature, methods employing the sequence form representation do not necessarily result in finding sequential equilibria.

Bibliography

[1] David Avis and Komei Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8:295–313, 1992.

[2] Patrick Bajari, Han Hong, and Stephen Ryan. Identification and estimation of discrete games of complete information. Manuscript, Duke University., 2004.

[3] Ben Blum, Christian Shelton, and Daphne Koller. Gametracer. Software, version 0.2, http://dags.stanford.edu/Games/gametracer.html, 2004.

[4] George Dantzig. A proof of the equivalence of the programming problem and the game problem. In Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 330–335. Wiley, 1951.

[5] Federico Echenique. Finding all equilibria. Caltech Social Science Working Paper 1153, 2003.

[6] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. The complexity of eliminating dominated strategies. *Mathematics of Operations Research*, 18:553–565, 1993.

[7] Jacob Goeree and Charles A. Holt. Ten little treasures of game theory, and ten intuitive contradictions. *American Economic Review*, 91:1402–1422, 2001.

[8] Srihari Govindan and Robert Wilson. A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, 110:65–86, 2003.

[9] Srihari Govindan and Robert Wilson. Computing Nash equilibria by iterated polymatrix approximation. *Journal of Economic Dynamics and Control*, 28:1229–1241, 2004.

[10] John C. Harsanyi and Reinhard Selten. *A General Theory of Equilibrium Selection in Games*. MIT Press, Cambridge MA, 1988.

[11] P. Jean-Jacques Herings and Ronald J.A.P. Peeters. A differentiable homotopy to compute Nash equilibria of n-person games. *Economic Theory*, 18:159–185, 2001.

[12] Kenneth L. Judd. Computational economics and economic theory: substitutes and complements? *Journal of Economic Dynamics and Control*, 21:907–942, 1997.

[13] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[14] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259, 1996.

[15] C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

[16] C. E. Lemke and Jr. Howson, J. T. Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12:413–423, 1964.

[17] Oscar Mangasarian. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12:778–780, 1964.

[18] Richard D. McKelvey. A Liapunov function for Nash equilibria. Caltech Social Science working paper, 1991.

[19] Richard D. McKelvey and Andrew M. McLennan. Computation of Equilibria in Finite Games. In Hans Amman, David A. Kendrick, and John Rust, editors, *The Handbook of Computational Economics*, volume I, pages 87–142. Elsevier, 1996.

[20] Richard D. McKelvey and Andrew M. McLennan. The maximal number of regular totally mixed Nash equilibria. *Journal of Economic Theory*, 72:411–425, 1997.

[21] Richard D. McKelvey and Thomas R. Palfrey. An experimental study of the centipede game. *Econometrica*, 60:803–836, 1992.

[22] Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10:6–38, 1995.

[23] Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1:9–41, 1998.

[24] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

[25] Ryan W. Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 2006.

[26] Rahul Savani and Bernhard von Stengel. Exponentially many steps for finding a Nash equilibrium. CDAM Research Report LSE-CDAM-2004-03, 2004.

[27] Herbert Scarf. The approximation of fixed points of a continuous mapping. *SIAM Journal of Applied Mathematics*, 15:1328–1343, 1967.

[28] Lloyd Shapley. A note on the Lemke-Howson algorithm. *Mathematical Programming Study*, 1:175–189, 1974.

[29] Theodore L. Turocy. A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence. *Games and Economic Behavior*. Forthcoming.

[30] G. van de Laan, A. J. J. Talman, and L. van Der Heyden. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Mathematics of Operations Research*, 12:377–397, 1987.

[31] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Annals of Mathematics*, 100:295–320, 1928.

[32] Bernhard von Stengel. Computing equilibria for two-person games. In Robert J. Aumann and Sergiu Hart, editors, *Handbook of Game Theory, With Economic Applications*, volume 3, pages 1723–1759. North-Holland, 2002.

[33] Robert Wilson. Computing simply stable equilibria. *Econometrica*, 60:1039–1070, 1992.

[34] Yoshitsugo Yamamoto. A path-following procedure to find a proper equilibrium of finite games. *International Journal of Game Theory*, 22:249–259, 1993.