Analyzing Computational Models[*]

David A. Siegel[†]

Abstract

Computational models have been underutilized as tools for formal theory development, closing off theoretical analysis of complex substantive scenarios that they would well serve. I argue that this occurs for two reasons, and provide resolutions for each. One, computational models generally do not employ the language or modes of analysis common to game-theoretic models, the status quo in the literature. I detail the types of insights typically derived from game-theoretic models and discuss analogues in computational modeling. Two, there are not widely established procedures for analysis of deductive computational models. I present a regularized method for deriving comparative statics from computational models that provides insights comparable to those arising from game-theoretic analyses. It also serves as a framework for building theoretically tractable computational models. Together, these contributions should enhance communication between models of social science and open up the toolkit of deductive computational modeling for theory-building to a broader audience.

[*] I thank Jon Bendor, Keith Krehbiel, Charlotte Lee, Scott de Marchi, William Minozzi, John Patty, and Ken Shotts, as well as Bill Jacoby and three anonymous reviewers at *AJPS*, for helpful comments.

[†] Associate Professor, Department of Political Science, Duke University; E-mail: david.siegel@duke.edu.

Computational modeling is an approach, complementary to game theory, in which one formally specifies rules of behavior, and then computes the consequences of these rules in a population of actors, often called agents (De Marchi and Page, 2014; Miller and Page, 2007). Its merits include improvements in our ability to: model large, finite, and heterogeneous populations; model complex interdependencies between actors and/or deviations from strict rationality assumptions; and explore system dynamics and non-equilibrium behavior (De Marchi and Page, 2014). Given these merits, computational modeling is arguably underutilized as a tool of formal modeling. I propose two reasons for this, and offer possible resolutions for each.

First, computational models generally do not employ the language or the modes of analysis common to game-theoretic models. This a problem given game theory's place as the status quo in developing formal political theory. I address this by detailing the types of insights typically derived from game-theoretic models and discussing analogues in deductive computational modeling. I focus particularly on comparative statics, as these often generate a model's testable hypotheses. In game-theoretic models, one derives comparative statics to illustrate the manner in which the equilibrium of the model changes with exogenous parameters. One can do the same in computational models, even those employing simulation-based approaches (replacing "equilibrium" with "steady-state"). My hope is that fostering a common language can spur more productive engagement between these complementary methodologies.

Second, there are not widely established procedures for deductive computational model analysis, as there are for game-theoretic models. To address this, I present a regularized, hierarchical procedure for building computational models and computing their comparative statics. There are three major benefits from following this approach, which I call sequential parameter sweeping (SPS). One, the computation of comparative statics provides directly comparable insights to those arising from a game-theoretical analysis. Two, the procedure I detail for the computation involves sequentially and hierarchically constructing a model in a manner designed to make it theoretically tractable. This provides a framework for model development and for building on previous work. Three, the procedure relies on techniques that are commonly known to empirical scholars, including simulation and numerical optimization, opening to them the toolkit of computational modeling for theory-development. I illustrate this last point with an example of a computational model intended to provide theoretical expectations for a program evaluation.

Analysis Machines

Broadly speaking, game-theoretic models are used to derive one of three types of insight. The first arises directly from the equilibrium of the model. The second arises from the strategic incentives present in equilibrium. The third arises from the comparative statics of the model. In this section I will discuss each in turn, along with its analogue in computational models.

Before doing so, I offer two scope conditions for my discussion. First, I leave more detailed positive arguments for the use of computational models to others; see De Marchi and Page (2014) for a recent example. Second, I limit my attention to deductive computational models, since game theory is a deductive enterprise and I am attempting to illuminate the commonalities between game-theoretic and computational modeling.[1]

Deductive computational models have many of the same characteristics as do game-theoretic models.[2] Specifically, one assumes a set of actors, actions, and outcomes from joint actions in each. In both cases these assumptions go into an analysis machine, and one turns its crank to produce insights.[3] In game theory that machine is joint utility optimization and the crank is algebra and analysis. In computational modeling (of the deductive sort)

[1] In the language of De Marchi and Page (2014) these are "intuition engines" (9), rather than data-driven "high-fidelity models." Though I confine discussion to models in which conclusions are drawn from a single set of assumptions, as in game theory, the method of analysis I propose could be folded into a generative modeling approach, which seeks systems of assumptions that could account for particular regularities (Epstein, 2006).

[2] I use the phrase "game-theoretic models" as shorthand for "game-theoretic models solved analytically." I class "game-theoretic models solved computationally" with other computational models for reasons given below. I do not significantly address other types of deductive formal models that are solved analytically (e.g., Bendor et al. 2010) as these are less common in political science. I also focus on actor-level behavior, rather than system dynamics, for comparability with game theory.

[3] Thanks go to Keith Krehbiel for the analogy. See Hunt et al. (2008) for a related argument in biology.

that machine is a set of behavioral rules dictating actions to be taken in response to stimuli and the crank is computation (which is really just arithmetic).[4] Each is more useful in some domains, but there is essentially no difference between the goals of the two forms of deductive modeling.

To make this clearer I will stick to the phrase “computational model” rather than the more common “agent-based model (ABM).” The former phrase is informative in the same way “game-theoretic model” is; it describes the machine (or set of possible machines) and the crank in use. The latter is uninformative: strictly speaking, game-theoretic models are also based on agents’ behavior, and utility maximization is a valid behavioral rule.[5]

Insights from Equilibrium/Steady States

Analyzing a game-theoretic model entails proving the existence of an equilibrium of the model, and often deriving it. Thus, I focus first on the importance of the equilibrium itself. From one perspective, the enormous literature analyzing simple games, such as those of cooperation, coordination, and public goods, centers on equilibrium outcomes. From cooperation games like the prisoner’s dilemma (PD), we learn that we can arrive at socially worse outcomes if individuals have incentives to deviate from socially better actions. From coordination games we learn the difficulty of coordinating on an outcome when more than one could be an equilibrium. From (continuous) public goods games we learn that public goods may be under-provisioned in equilibrium. What these examples, and many more, have in common is that they are derived from the point prediction produced by game-theoretic analysis: the equilibrium strategies played by each actor.

[4] I do not limit the definition of behavioral rules to those with psychological origins. They may include, for example, evolutionary mechanisms of selection, replication, and mutation; rational choice; or other rules that dictate responses to stimuli.

[5] The term “ABM,” as a synonym for individual-based model, is more useful when intended to contrast with population- or system-level analysis. This distinction is about unit, rather than method, of analysis, though; computational models may be used both at the individual and the system level. Given the dearth of population-level formal theorizing in political science, there is little reason for unit-level distinctions. In fact, I believe that, rather than usefully delineating differences between the approaches, calling the use of computers to assess models “agent-based modeling” only serves to artificially separate approaches and make it easier to discount computational modeling entirely.

The analogue to this in computational models is analysis of the steady state or limiting distribution of the model. Because computational models can take many forms, this claim needs unpacking. Some computational models, e.g., Baron and Herron (2003); Groseclose (2001), are virtually identical to game-theoretic ones. The behavioral rule they employ is utility optimization, and the computer simply numerically computes the equilibrium of the encoded game. In these cases the steady state of the computational model is an equilibrium that is exactly equivalent to that found in game-theoretic models, and admits identical insights.

Other computational models, e.g., Bendor et al. (2011); Kollman et al. (1992), may employ utility functions or similar preference constructs for actors, but do not assume that actors optimize according to their preferences. Rather, actors in these models utilize other behavioral rules to choose actions, such as satisficing or hill-climbing. Still other computational models, e.g., Laver (2005), do not specify formal preferences at all, instead assuming behavioral rules directly. Computational models of either type do not produce outcomes exactly equivalent to game-theoretic equilibrium; there is no equivalent requirement in such models to the game-theoretic notion of no beneficial deviation by any actor. However, the steady states and limiting distributions of these models are still directly analogous to game-theoretic equilibrium, allowing these types of computational models to produce similar insights to game-theoretic ones.

To see how, note that the appropriate language for such computational models is that of the stochastic process (Laver and Sergenti, 2011).[6] Each of these models has a state vector that specifies the core properties of the system at any time.[7] The actions of the actors, as dictated by their behavioral rules, contribute to a transition rule that takes the system between its states. In other words, transitions between states in the stochastic

[6] A deterministic model is a degenerate stochastic process. Stochastic processes are formal mathematical constructs that can admit analytic solutions, depending on the particular process modeled. In other words, not all models of stochastic processes need be solved computationally. See Bendor et al. (2010) for an example.

[7] The use of a computer forces the number of states to be finite, though potentially effectively infinitely large.

process are functions of prior states and present and prior actions. These transitions may be deterministic or they may be stochastic. A steady state or limiting distribution represents the state or distribution over states that is invariant under the transition rule. These notions thus represent the equilibrium of the stochastic process in the truest sense of the word, as they characterize the distribution of states that, once reached, is henceforth unchanging.

If the stochastic process encoded by a particular computational model is *ergodic*, then it converges to its unique limiting distribution from any initial distribution over states. This unique limiting distribution is just like the unique equilibrium of a game-theoretic model, and can be interpreted similarly. In both cases the model produces a singular outcome that, once reached, will henceforth be unchanging. If this outcome depends on the parameters of the model, then, as I discuss below, we can compute comparative statics to discern how the outcome changes with the parameters. From this analysis we can draw hypotheses about the dependence of the equilibrium on the parameters.

If the stochastic process encoded by a computational model is not ergodic it may have more than one steady state. An example of this would be a system with multiple absorbing states, as is common in the presence of path dependence (Page, 2006). In such cases we don't know in which state the system might end up, though, at least in the presence of multiple absorbing states, we often know that the system will end up in one of these states. This is analogous to multiple equilibria in a game theoretic model, and has all the same problems of multiple equilibria: it is unclear how to interpret outcomes and how to assess comparative statics. That said, we can address these problems in the same fashion as we do in the presence of multiple equilibria. We can either eliminate unwanted equilibria via refinements or derive comparative statics that apply to more than one equilibrium/steady state/limiting distribution, as is done, for example, in monotone comparative statics (Ashworth and Bueno de Mesquita, 2006).

Regardless of the number of different steady states, however, the central point is that a steady state or limiting distribution may be interpreted just as is an equilibrium in game theory: as the point from which no actor will alter the distribution of actions it is taking.[8] Computational modelers seeking to compare the intuition behind their results to that derived from game-theoretic equilibrium would do well to derive steady states or limiting distributions from their models. Framing their results accordingly will enhance readers' ability to assess the intuitions derived from their models and will reduce confusion stemming from a lack of comparability. Doing so will also highlight the mathematical rigor of the method: the steady state or limiting distribution of an ergodic process, for instance, is a well-defined mathematical object, as is an absorbing state.

Further, deriving steady states or limiting distributions will allow computational models to engage in deriving theories of institutions, rather than solely institutional analysis (Diermeier and Krehbiel, 2003). In other words, it will allow one to address the question, via comparison of steady states, of which institutions produce normatively better outcomes, as game-theoretic work has done with respect to institutions such as judicial review (Dragu and Board, 2015; Fox and Stephenson, 2011).

Insights from Strategic Incentives/Pathwise Properties

Game theory's focus may be on equilibrium behavior; however, underlying every equilibrium are the strategic incentives of the actors, and these incentives can be directly of interest to us. Consider again the PD. The equilibrium of mutual defection is not the only object of interest; we also learn from the existence of a private incentive to defect from cooperation. Or consider a public goods game. We care not only about the equilibrium of sub-optimal provision, but also the incentive to free-ride off others' contributions. Though these incentives are often viewed as part and parcel of an equilibrium, they are in many ways more general. For example, one might have an incentive to defect or free-ride even if one's behavioral rule is not optimization, and even if no equilibrium is reached.

To get at the incentives directly in a game-theoretic context, one looks to the derivation of the equilibrium. Best response functions provide the clearest connection to incentives, as

[8] This assumes, of course, that actions are part of the state.

they tell the analyst exactly how an actor optimally responds both to other actors' actions and to exogenous parameters. In other words, best response functions express incentives to take actions given specific contexts.[9] For example, the best response of each actor in a PD to all actions by one's opponent is to defect, capturing the private incentives that lead inexorably to the unique equilibrium of mutual defection.

Behavioral rules encode these incentives in much the same way as do best response functions. A behavioral rule dictates how an actor should respond to a particular set of stimuli. If that response is optimization and the stimuli are other actors' actions and exogenous parameters, then the behavioral rule and the best response function are *identical*.

When responses and stimuli in behavioral rules differ from those in game theory, we can still derive an analogue to incentives. Here we use the concept of pathwise properties in stochastic processes. If a steady state is where a stochastic process ends up, the path of a process is the way it gets there. Pathwise properties are characteristics that all paths betray. Importantly, one can specify the path (or paths) the system takes when there are many steady states, or even none at all. For example, Bendor et al. (2011) model retrospective voting via a stochastic process in which a voter becomes more likely, over a series of elections, to vote for a party the better that party has performed for that voter in the past, according to that voter's interests. Steady-state properties of this model speak to long-run distributions of voter behavior in the population. Pathwise properties, in contrast, specify the manner in which the distribution of voters' behavior changes from election to election, in this case moving toward the steady state.

Pathwise properties thus connect directly to incentives, in that they express the underlying impulses of actors that drive their behavior over time, and in some cases produce a steady state. To make this analogy more concrete, consider the simple example of a participation game with strategic complementarities and a constant cost of participation. Though the

[9] In the absence of best response functions, one can discuss trade-offs that represent dueling incentives. The technique of monotone comparative statics particularly highlights this (Ashworth and Bueno de Mesquita, 2006), as it deals with comparisons of marginal gains and losses.

equilibrium of such a game is static, the incentives that underlie it produce a best response function that is increasing in the number of others who are already participating. A simple behavioral rule in which others' participation increases one's own likelihood of participating encodes similar incentives. The pathwise properties of a stochastic process based on such a rule will capture these incentives dynamically. For example, beginning from a point of no participation in the population, along all paths participation of all individuals will weakly increase as people over time respond to others' choices. Though in this example the incentives are obviously hardwired into both types of models in a way that does not require solving the game or deriving pathwise properties to identify, the point remains for more complex situations: pathwise properties dynamically capture behavioral regularities that mirror incentives in game theory.

In short, then, computational modelers seeking to compare the intuition behind their results to that derived from game-theoretic incentives would do well to derive pathwise properties from their models. Doing so would allow them to compare changes in behavior over time in their models to the incentives that drive equilibrium behavior in game-theoretic models.

Insights from Comparative Statics

Equilibrium predictions in game-theoretic models present a high bar for empirical testing. They are point predictions, after all, and in the social sciences we rarely believe our models are sufficiently well-determined so as to produce accurate point predictions. Thus, particularly for models in the tradition of EITM—Empirical Implications of Theoretical Models—we often focus not on the equilibrium of the model, but rather on the manner in which the equilibrium changes with exogenous parameters. We call this deriving the comparative statics of the model, because we are comparing different static equilibria that arise with different values of the parameters.

Various techniques exist for determining the comparative statics of a game-theoretic model, from simple differentiation of an explicitly solved equilibrium, to implicit differentiation of an expression that defines an equilibrium, to techniques like monotone comparative statics which can produce comparative statics without necessarily solving for the equilibrium first (Ashworth and Bueno de Mesquita, 2006). In all cases the result is the same: a statement of how an endogenously determined outcome of the model changes with an exogenous parameter.

Comparative statics can also be derived from any computational model that admits a steady state or limiting distribution.[10] They have the exact same meaning as do those in game-theoretic models, and allow direct comparisons. Computational modelers seeking to compare the intuition behind their results to that derived from game-theoretic comparative statics would do well to derive comparative statics from their models. I suggest procedures for how to do so in the next section. Following these procedures as one builds a computational model has added benefits as well. One, doing so helps ensure that one's computational model is theoretically tractable, meaning that one can derive clear causal linkages between endogenous variables and exogenous parameters from an analysis of the model. Two, doing so helps guide researchers looking to build upon previous work by defining the scope of what can be added without losing sight of causal linkages. Before getting to this, though, I briefly elaborate on the connections between computation and analysis.

Computation Versus Analysis

I summarize the connections between game-theoretic and computational modeling in Table 1. The key distinction that has been highlighted thus far is the "crank" row. As we'll see in more detail in the next section, computation opens up new kinds of models to rigorous analysis but cannot generate closed-form solutions. This makes certain kinds of analysis, such as comparative statics, trickier in all but the simplest cases.

Though the "crank" turned in our analysis machine is often correlated with the type of machine, it need not be. While most examples of formal modeling in political science

[10] Again, as above, finding comparative statics when there are multiple potential steady states produces the same problems as does finding them when there are multiple equilibria.

Type of Model:	Game Theory	Computational Modeling
Analysis Machine:	Joint Utility Optimization	Any Set of Behavioral Rules
"Crank":	Mathematical Analysis	Computation
Long-run Solution Concept:	Equilibrium	Steady State/Limiting Distribution
Incentives Visible in:	Best Responses	Pathwise Properties
Effect of Parameters via:	Comparative Statics	Comparative Statics

Table 1: Modeling Connections

are either analytic game-theory or computational modeling of boundedly rational behavior, as previously noted one could solve models of boundedly rational behavior analytically or game-theoretic models computationally.

Comparative Statics from Computational Models

There are numerous ways to derive useful results from computational models. The method I describe here has a specific intent: to provide comparative statics that are comparable to those derived in game-theoretic models. This method is most useful when employing computational modeling as a deductive theoretical enterprise, and may be less useful in other contexts. The foundation of the method is nested, hierarchical modeling: first one develops and analyzes a simpler model than one wants, then one attempts to add complexity in manageable stages. This foundation bears similarity to common statistical practice, as well as the common practice in applied game theory of starting with a well-understood model and then adding one or more new wrinkles. It also is related to the idea of a nearly decomposable system Simon and Ando (1961).

The Problem

Before describing the method, it helps to identify the challenges in deriving comparative statics from computational models. Comparative statics of a game-theoretic model are mathematical statements specifying the manner in which endogenous equilibrium variables change with variation in exogenous parameters. Most commonly, they are expressed as a set of derivatives. Because these are derived analytically, one can deduce the manner in which interactions of parameters affect the equilibrium. For example, if an equilibrium variable, $x^*(\alpha, \beta)$, is a function of its two parameters, then the comparative static $\frac{dx^*(\alpha,\beta)}{d\alpha}$ can be a function of both parameters as well. From this one can discern not only if x^* is increasing in α, but also if this relationship depends on the value of β.

Mirroring this flexibility presents a problem for computational modeling. To make this problem as stark as possible, consider a behavioral rule of joint utility optimization, so that the computational model computes the game-theoretic equilibrium numerically. How would one compute the comparative statics of this model?

Let's start with a single parameter and so a one-dimensional parameter space. Ideally we want the derivative of the equilibrium at every point in this space. In practice we can get away with computing it at a finite subset of points as long as they are sufficiently close together and the equilibrium does not respond too quickly to its parameter. Assume that we subdivide the parameter space into regions of width ϵ, where ϵ is small relative to the rate of change of the equilibrium in its parameter. If we then compute the value of the equilibrium at the endpoints of these regions, we can linearly approximate comparative statics at any point in a straightforward manner: $\frac{dx^*(\alpha)}{d\alpha} \approx \frac{x^*(\alpha+\epsilon)-x^*(\alpha-\epsilon)}{2\epsilon}$.[11]

If there are instead two parameters, one could subdivide the two-dimensional parameter space into a grid with squares of area ϵ^2 and compute the derivative in each dimension, at the cost of an exponential increase in computational time. If this is not a barrier, one could even compute the cross-partial derivative at each grid point in order to understand the pattern of interaction between the parameters. This requires performing the same differencing operation used to compute the derivative with respect to the first parameter, but this time on the derivative itself with respect to the second parameter: $\frac{\partial^2 x^*(\alpha,\beta)}{\partial\alpha\partial\beta} \approx \frac{\frac{\partial x^*(\alpha,\beta+\epsilon)}{\partial\alpha} - \frac{\partial x^*(\alpha,\beta-\epsilon)}{\partial\alpha}}{2\epsilon}$.

In theory, this process can be continued up to any number of parameters and any di-

[11] There are other, higher-order approximations of the derivative one could compute, as well as rules of thumb for the optimal value of ϵ to use. But we are just concerned with the existence of a computed derivative.

mension of interaction among the parameters. If the computation of the equilibrium is straightforward, the equilibrium well-behaved, the domain of each parameter bounded, and the number of parameters small, this may be the way to go. It certainly provides the most complete comparative statics.

However, if any of these conditions fails to hold, then the exponential increase in computational time will constrain and this procedure becomes untenable. This is one manifestation of the "curse of dimensionality" (De Marchi, 2005) and it underlies the common computational modeling maxim KISS: keep it simple, stupid (Axelrod, 1997). But what if modeling the interaction you want to model requires more than a couple of parameters?

A commonly proposed solution to this problem is the use of sampling. For example, Smirnov and Fowler (2007) sample parameter values uniformly across the parameter space and compute equilibria and their first derivatives at each sampled point. This is a time-efficient procedure given a slightly larger set of parameters and may be very effective when the equilibrium of the model is well-behaved.[12]

But what if the model is not well-behaved, perhaps because it is expected to produce non-linearities or non-monotonicities, as in many computational models? Then one must potentially substantially oversample in regions of the parameter space in which the equilibrium displays non-linearities or non-monotonicities. This is difficult to do when one does not start knowing the location of such regions. The same sampling problem holds if parameters have potentially unknown interactive effects: one must oversample appropriate regions of the parameter space that capture important parameter combinations. Thus, if the model is not well-behaved, an approach based on sampling becomes problematic and is not likely to reproduce the range of insights from analytic comparative statics, though it can be useful as a first cut and in conjunction with other methods.

[12] Smirnov and Fowler (2007) assume that "the parameters [of their model] are drawn equally from all parts of the space and each point in the space has an equal chance of contradicting the claim [of a particularly signed comparative static]" (17) to argue that one can bound one's confidence in one's direct computation of a comparative static. They compute derivatives along one direction at a time numerically in a manner similar to that given above.

The sampling problem gets worse when, rather than computing derivatives at each point, one instead regresses equilibrium variables on exogenous parameters (e.g., Kim and Bearman 1997). This more common procedure adds regression model misspecification to the sampling problem. If the computational model instantiates a data-generating process (DGP) in which non-linearities, non-monotonicites, and varying interactive effects are produced, then a proper specification of the regression model should include these. However, unless one knows ahead of time the exact nature of all of these complications, the regression model will have to include everything, making it extremely difficult to interpret. Leaving pieces out, though, risks bias.[13]

Sequential Parameter Sweeping

The issue, then, is that any approach that mitigates the problem of having too many parameters to fully compute comparative statics risks missing the sort of complexity that led one to employ a computational model in the first place. As a result, these approaches do not guarantee analogous insight to that produced by analytic comparative statics of game-theoretic models, though they can get close when the equilibrium is well-behaved.

To accommodate the computation of more comparable comparative statics for a wider range of computational models, I offer an alternative approach: *sequential parameter sweeping (SPS)*. The approach is relatively straightforward and employs a simple idea. To fully analyze a more complex model, first analyze simpler versions of the model and build up from there. In other words, utilize a nested, hierarchical model structure. This idea is quite commonly employed in both empirical and game-theoretic modeling. In empirical modeling, one often begins with simpler models that contain only key variables before exploring more complex ones. In game-theoretic modeling, much applied modeling begins either by: (1) starting with a well-understood model and adding one or more aspects central to a new

[13] In fairness, if the true DGP is actually mirrored well by the computational model's DGP, but the statistical model used to test the theory betrays the same misspecification as the one used in deriving comparative statics from the computational model, then one is doing nothing worse in one's computational model than in one's empirical model. However, I know no one who would argue that the best way to accommodate a complex DGP statistically is to misspecify one's theory!

theory, such as commonly observed, for example, in models of electoral dynamics built upon spatial models; or (2) writing down one's ideal model and then eliminating inessential factors that lead to intractability. In all cases, this is done in order to better understand the operation of key parameters without problems of interpretation generated by complications.

The general SPS approach operates similarly. Below I describe SPS informally, using a simple game-theoretic example to highlight its core assumptions. A second, more complex computational example follows to illustrate the use of SPS in practice. An online appendix offers a more formal presentation of the method.[14] The flowchart in Figure 1 summarizes the steps in SPS and may be a useful reference.

SPS Informally

The first thing one does when using SPS to analyze one's model is to identify and fully analyze, via the computation of both steady state and comparative statics, a simplified version of one's model that can be captured in only 1-3 parameters. This serves as step "zero" in SPS.

We'll call the full model one wants to analyze M, and this simplified model M_0. A good M_0 does not need to be sufficient to answer one's research question. In fact, if it were, one would not need SPS. Rather, M_0 should be a model that answers a more basic research question, one which must be understood fully before turning to more complex questions. For example, before understanding mobilization under repression, one needs to understand mobilization in its absence. Before understanding coordination in complex informational environments, one must understand it in simple environments. Before understanding the effects of legislative institutions on bargaining, one must understand bargaining dynamics in the absence of institutions. And so on.

The key is to identify in one's model a core dynamic that is theoretically distinct on its own. Then one forms M_0 from only the parameters essential in capturing it.[15] Though

[14] The appendix may be found at `http://people.duke.edu/~das76/Research/Siegel_Analysis_AJPS_appendix.pdf`.

[15] Boiling down one's model to its most essential components is also a good practice in general, even if

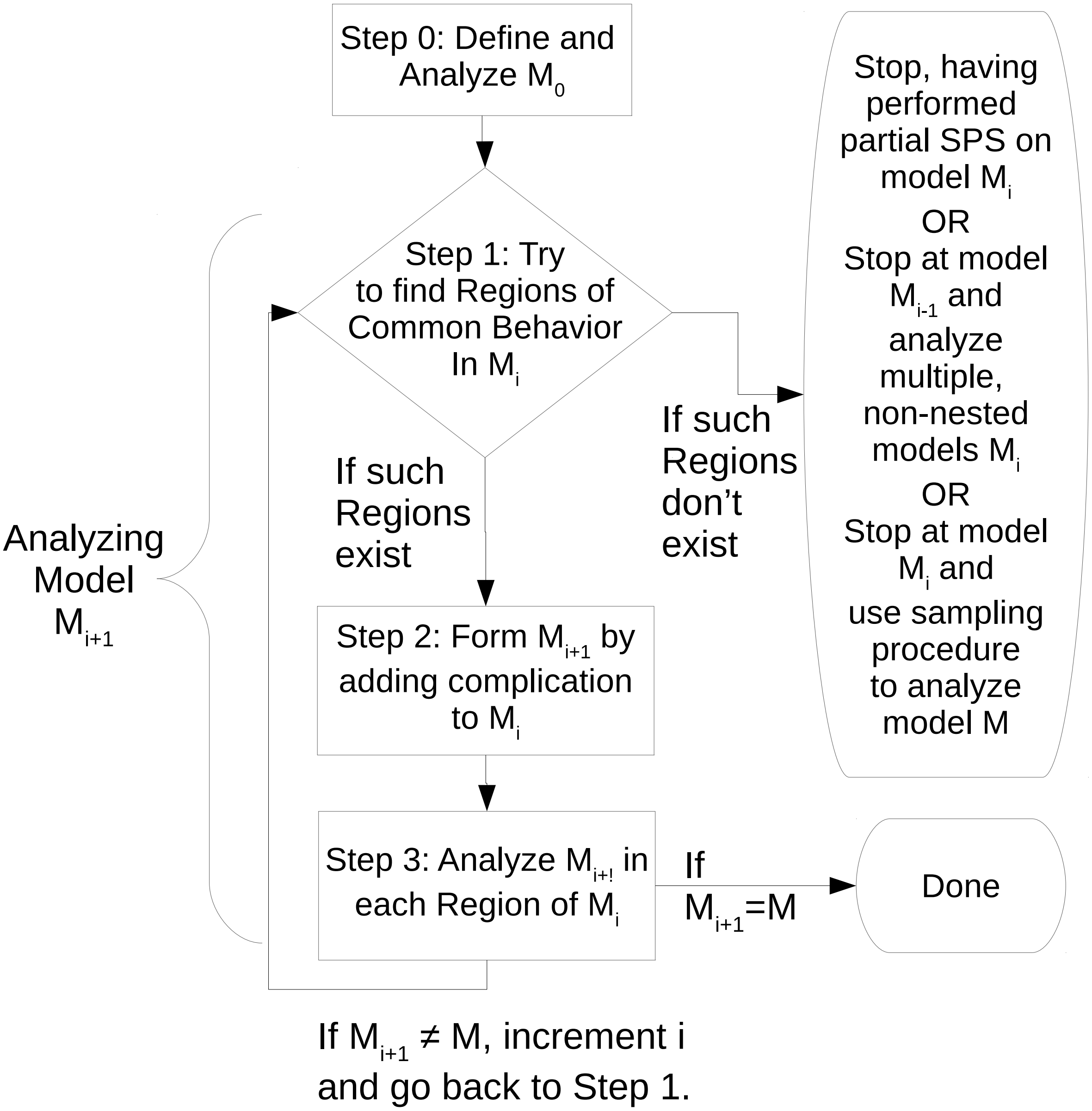
Step 0: Define and Analyze M_0
Step 1: Try to find Regions of Common Behavior In M_i
If such Regions exist
If such Regions don't exist
Stop, having performed partial SPS on model M_i
OR
Stop at model M_{i-1} and analyze multiple, non-nested models M_i
OR
Stop at model M_i and use sampling procedure to analyze model M
Analyzing Model M_{i+1}
Step 2: Form M_{i+1} by adding complication to M_i
Step 3: Analyze $M_{i+!}$ in each Region of M_i
If M_{i+1}=M
Done
If $M_{i+1} \neq$ M, increment i and go back to Step 1.

Figure 1: Flowchart for SPS

it is always possible to choose 1-3 parameters from a model and build a simplified model around them, focusing on theoretically distinct core dynamics helps ensure one's choice of simplified model is meaningful, which also aids in the successful application of SPS. In other words, substantive considerations should drive the choice of M_0 when one is not uniquely determined by the research question.

Analyzing M_0 presents no problem by construction. One needs only to compute the steady state at sufficiently fine-grained points in the 1-3 dimensional parameter space and then compute derivatives at each point. Simply plotting the steady state with respect to these parameters (using conditional plots) is often sufficient to discern the signs and behavior of the comparative statics at each point, and can be a good way to display results. The second example below illustrates this presentation style.

Let's see how finding an M_0 could work in practice, using a deliberately simple game-theoretic model as a template. The setting for the example is interpersonal trade. Specifically, the goal is to capture the level of trade between two people conditional on: (i) personal endowments and (ii) a context in which the people might already be engaging in cooperative or non-cooperative activities. Though one could construct a quite complicated model to capture this setting, its description is intended to suggest a simpler, nested solution in which a model of interpersonal trade (M) is layered on top of a model of cooperation (M_0). Though this M_0 is insufficient to capture a research question revolving around trade as a function of cooperation, it can capture the logically prior question of when cooperation occurs,[16] and so is a good choice.

I choose a repeated PD to model cooperation. As its solution does not require computation and will be familiar to many readers, it allows a tighter focus on the issue of model separability that will aid in understanding how the approach might work for more complex models.

one ends up elaborating on one's model later. This process is often necessary in game-theoretic models to get something tractable. For example, an unparameterized linear utility in a game-theoretic model may be perfectly sufficient to capture the effect of signaling, while keeping the model tractable.

[16] The question of cooperation is logically prior because trade is assumed conditional on cooperation.

A general repeated PD requires five parameters for each actor: four for payoffs and one for the discount factor. That's a total of ten parameters for two players, which is a significant number. Most are not necessary to capture the underlying dynamic, however. I use the PD from Fearon and Laitin (1996) and elide the in- and out-group complications in their model. This leads to a simple model M_0 that has three parameters which are identical for both types of actor: one to capture the temptation payoff from defecting against a cooperator, α, one to capture the sucker's penalty of cooperating with a defector, β, and one to capture the discount factor, δ. Its payoff matrix for the stage game is:

$$\begin{array}{|c|c|c|} \hline & \text{C} & \text{D} \\ \hline \text{C} & 1{,}1 & -\beta, \alpha \\ \hline \text{D} & \alpha, -\beta & 0{,}0 \\ \hline \end{array}.$$

To analyze this game we'll posit an equilibrium/steady state maintained by the use of a grim trigger strategy.[17] Given this punishment strategy, mutual cooperation can be maintained as long as $\delta \geq \frac{\alpha - 1}{\alpha - 0}$.

Having identified an M_0, we now turn to the three-step iteration that forms the core of the SPS approach. The goal of each iteration is to construct and fully analyze one of a series of nested models indexed by i, M_i, such that $M_m = M$ for some finite m. The goal of the first step of the iteration is to identify regions of the parameter space that behave similarly under the action of the model. This step is first applied to M_0, and later to any more complex M_i short of the full model M.

More explicitly, step one of SPS requires that the analyst see if each M_i is *separable into regions of common behavior*. I provide a more formal definition of the italicized phrase in the appendix. Intuitively, though, it means that one can *label* regions of the parameter space of M_0. Within each labeled region comparative statics with respect to all exogenous parameters in M_0 have roughly the same shape. The phrase "roughly the same shape" is a measure of numerical tolerance, as discussed in the appendix. It suggests that the qualitative behavior of the model, if not the point predictions of the model, is common to all sets of parameter

[17] Equilibria in repeated games are particularly easy to model computationally as they consist of rules specifying behavior under all circumstances of prior play. This is the very definition of a behavioral rule.

values that lie within a given labeled region. Further, not only does this common behavior not change much as the model is made more complex, but comparative statics for exogenous parameters that arise in more complex models also largely depend only on these labels, and not on the explicit values of the exogenous parameters in M_0.

Let's return to the interpersonal trade setting to elaborate on this notion. The equilibrium conditions we found in our analysis of M_0 specified two distinct regions of the parameter space: a cooperative region occurring whenever $\delta \geq \frac{\alpha-1}{\alpha-0}$ holds and a non-cooperative region occurring whenever $\delta < \frac{\alpha-1}{\alpha-0}$ holds. Are these regions of common behavior in the manner just specified?

If they are, then within each region two things are true. One, the comparative statics with respect to all parameters in M_0 should have roughly the same shape within each region. This is easily verifiable in this case. Variation in α and δ changes the location of the boundary between these regions, but does not change the value of any endogenous variable (here the actors' strategies) *within* a region. Further, variation in β has no effect on either the location of the boundary between regions or the value of any endogenous model variable within a region.

Two, the comparative statics with respect to all parameters not in M_0 but in more complex M_i should roughly depend only on the regions of the parameter space of M_0, and not on the precise parameters that define the regions. Again this is true in this case, though this time by assumption: we have assumed that interpersonal trade (and so the comparative statics with respect to all new parameters in M_1, defined below) is conditional only on the presence or absence of cooperation, and not on the parameters (i.e., not on α, β, and δ) that lead to each form of behavior. Consequently, the cooperative and non-cooperative regions of the parameter space of M_0 represent regions of common behavior in model M_0.[18]

[18] We could have gone further than this with respect to β. Since β acts completely independently of α and δ, we could have specified an even simpler model M_0 that did not include β at all and still captured cooperation, at least under grim trigger. We could then have constructed the more complex model out of M_0 and β by adding β to the payoff matrix, as above. This new model would equally well be separable into regions of common behavior.

Step one is the heart of SPS, and separability represents both a weak and a strong assumption. It's weak in the sense that it does *not* require that the equilibrium/steady state/limiting distribution be *identical* for all parameter values within a region. It may very well be, for example, that the level of interpersonal trade in our example is higher within the cooperative region in equilibrium when $\alpha = 2$ than when $\alpha = 3$. In this way it is weaker than the concept of a basin of attraction, which has the property that all parameter values within it lead to convergence of the model to some equilibrium/steady state/limiting distribution.

Separability is a strong assumption, though, in that it requires a certain level of independence between nested models. Specifically, the parameters of model M_0, such as δ, cannot interact with any parameters in the full model M that are not also in M_0 so strongly as to change the shape of the comparative statics for the latter parameters. More formally, assume, for instance, that r_1 is a parameter in the M_1 we will soon formulate for our interpersonal trade example, and x^* the equilibrium level of trade. Separability would require that $\frac{\partial^2 x^*}{\partial \delta \partial r_1}$ be small in the interiors of each of our two regions.

Due to the strength of this assumption—it is not always possible to label regions of the parameter space and have the comparative statics for more complex models depend only on these labels—separability as I have defined it does not always hold. Even when it does, identifying regions of common behavior can require substantial additional analysis.

That said, it is important to note that one need not rely only on the analysis of one's own simplified models to discern separability; one can also use extant theory. For example, one might build one's model, as I have done here, on an existing model which prior analysis has indicated possesses a small number of regions of common behavior. If one is careful to specify additional parameters that interact in their effects only with the type of region and not with the parameters that distinguish these regions (i.e., if one builds one's model hierarchically), then one can use this approach as well. In this way, the SPS approach becomes very similar to common model-building tactics in applied game-theoretic and regression analysis, as noted.

If such model construction is infeasible and step one cannot be completed, one must halt SPS. I discuss the consequences of this below. If one can complete step one, however, then one adds in step two a desired, self-contained complication to the model, as long as it is specified by no more than 1-2 new parameters.

We call this new model M_1. Choosing what complication to add first should be driven by substantive and logical considerations, much as was the choice of M_0. If there are remaining components of M that are still logically prior to others, these should be added first, as should components that yield the clearest message. If SPS will fail at some point this is particularly an issue: one may want to apply SPS to part of the model, as I discuss below, and the order of the added complications will matter for what parts of the model are analyzed by SPS. But, if none of these are at issue, the order of complications added may not matter at all.

Returning to our example, I assumed that trade was conditional on both cooperation and personal endowments. Thus, I define model M_1 as follows. Each actor, $k \in \{1, 2\}$, has an endowment parameter, r_k. The state of cooperation or non-cooperation is determined by a repeated PD that occurs in parallel with the trade decision. The latter depends on the presence or absence of cooperation and both endowments. Employing a function $C(\alpha, \beta, \gamma)$ which equals 1 under cooperation and 0 under non-cooperation, we thus can specify equilibrium trade as $x^*(C, r_1, r_2)$. In this case we have perfect separability, in that $\frac{\partial^2 x*}{\partial \nu \partial r_k} = 0$ for $k \in \{1, 2\}$ and $\nu \in \{\alpha, \beta, \delta\}$ within each region, and our earlier step one holds.

With the more complex model M_1 in hand, in step three we analyze it in the same manner previously described. Now we see the reason for the constraint of 1-2 additional parameters: adding two parameters effectively creates a three-dimensional parameter space. In undertaking this analysis, one uses representative values of the parameters of M_0 in each region of common behavior (i.e., each labeled region). Values are chosen for substantive reasons; their exact values do not significantly matter by construction. This produces comparative statics for the more complex model M_1.

For our example, this means choosing a set of representative values of α, β, and δ for each of our two regions identified in step one and then computing comparative statics with respect to r_1 and r_2 at each of these two sets of representative values.

As specified, our example is now complete: $M_1 = M$. The value of SPS in analyzing this example is two-fold. One, rather than having to deal with a five-dimensional parameter space in our full model, we only needed to analyze two two-dimensional parameter spaces. This is, of course, much more valuable for a true computational model than for our game-theoretic one, and we will see this in the example below.

Two, the necessity of applying SPS led us to construct a nested, hierarchical formal model to address our research question. This allowed the derivation of insight not just from the full model, but also from the nested cooperative model. This insight was trivial given our familiarity with the iterated PD, but would not have been for a more novel model.

Though we have now fully computed comparative statics for our example, SPS need not end after one pass for more complex models. The SPS approach dictates that one iterates steps one through three, switching M_i for M_{i+1}, until either one has fully analyzed one's desired model M or no regions of common behavior can be identified in step one. Depending on the flexibility one has in modeling, once can reduce the possibility of the latter by designing one's model complications to produce separable regions of common behavior. This is more likely when one's theory is itself structured hierarchically, or is comprised of many independent components.

To see how this might work, again return to our interpersonal trade example. Assume first that the analyst desires to study several behaviors in and out of a state of underlying cooperation, but that these behaviors are all independent, apart from their joint reliance on cooperation. In a case like this, one could continue to add complications to the model, iterating SPS with each complication. There are two reasons one can do this. First, no specific complication would alter the separability of M_0 into regions of common behavior. Second, if each model complication were independent of all others, than all models arising from these complications would remain separable as well.

This example displayed perfect separability between model components, but this is not necessary to apply SPS, as we will see in the computational example below. In general, as described in more depth in the appendix, a greater tolerance for numerical approximation allows more flexibility. The standard given as part of SPS focuses on rough similarities in the shapes of comparative statics within each region of common behavior, without a focus on the boundaries between regions, but this standard could be varied by end-users of the method based on need.

What one cannot have while performing SPS are comparative statics that depend strongly not just on rough properties (like cooperation) of regions of common behavior of simpler models, but also on specific parameter values of those simpler models. So, for example, we would not have been able to complete step one, above, had interpersonal trade depended not just on endowments in our M_1, but also strongly on the product of endowments and α and δ. While such a model might be a way to capture a particular type of psychological incentive for graft, the shape of the comparative statics for all endogenous variables in M_1 would depend in this case on the precise values of α and δ, giving us a four-dimensional parameter space to analyze.

When SPS Fails

We have seen that complex models can be analyzed via SPS, as long as they are developed deliberately in a hierarchical fashion or admit a hierarchical, nested decomposition. Thus, if one follows this approach and seeks out separable complications, one can develop a model that will yield by construction comparative statics providing the same sorts of insights as do analytic comparative statics arising from game-theoretic models.

Of course, not all models can be built or decomposed in this way. It might simply be the case that a model requires too many interactions between the parameters to be separable in the manner the method dictates, as in the variant M_1 capturing graft just described. One solution is to simplify the model to the point where SPS can be used. Even if one does not in the end go this route, the act of reasoning through whether this simplification is possible is useful in itself. If nothing else it provides supporting justification for one's model that will be important to include to head off claims that the model is unnecessarily complicated.

Should one not be able to simplify the model—and, again, one should be careful making this claim—then one has a few options. First, one can take SPS as far as it will go, and simply stop there. This leaves the full model unanalyzed, but does provide a clear picture of some simpler model M_k.

Second, one could take SPS almost as far as it will go and then apply multiple, non-nested extensions. To see how this would work, assume that one can iterate in SPS only up to some model M_k, leaving a series of other desired complications unanalyzed. In this case, one could return to model M_{k-1} and analyze a series of versions of M_k, each with a different complication added. This approach trades off an understanding of the interactions between all these complications for the ability to analyze each of them with respect to the more basic model M_{k-1}. This is effectively the approach typically taken in game-theoretic analysis, across different models' extensions of some base model, such as alternating-offer bargaining or spatial competition.

Third, again assuming one can iterate in SPS only up to some model M_k, one could analyze more complex models via a sampling procedure. As noted above, while this will not provide analogous comparative statics as would a full SPS approach, it is an improvement over solely sampling as it provides comparative statics for the simpler model. These may themselves be substantively interesting, and may yield insight into the behavior of aspects of the more complex model.

While these avenues exist for partial applications of SPS in order to build causal understanding within core aspects of the model, it is possible that deductive modeling is not the appropriate tool of analysis for irreducibly complex models. With so many necessary, interacting parameters in play, causal understanding is likely to be reduced over that available in a simpler model. The analyst may very well be better off with a more empirically-driven predictive model, accepting that some causal arguments are simply beyond our present ability to analyze. Even here, though, the act of attempting to perform SPS provides insight into the best tools with which to approach the problem.

A Computational Example

The interpersonal trade example illustrated SPS and the concept of separability, but as it did not show how SPS would work in practice for a true computational model, I build and analyze a simple one here.[19]

The model's setting is akin to program evaluation, save prospective rather than retrospective. I assume that government: (i) faces a heterogeneously dissatisfied population whose members interdependently consider participating in action against it; (ii) knows the size of the population and has some assessment of average dissatisfaction; (iii) can uniformly apply some economic or political intervention, such as infrastructure building or governance improvements, designed to reduce dissatisfaction; (iv) lacks information on levels of individual dissatisfaction (Kuran, 1991) and so cannot target this intervention at the most dissatisfied; and (v) is considering a program that would add a costly bureaucratic layer between government and population to gather information that would allow targeted interventions. The question is simple: How well would this information-gathering bureaucracy work, and is the program worth its cost? The answer would not only aid government planning, but also help us understand the causal mechanisms underlying the results of later empirical program evaluations, a necessary step in generalizing these to a larger class of potential programs and substantive settings.

To begin, we must first specify an M_0 that is a simplified version of our desired model and captures a logically prior theoretical question. The simplest such M_0 is a model of interdependent decision-making in the absence of government action. To make use of prior theoretical results I build M_0 out of a simplified version of the model in Siegel (2009), which has similar properties to Kuran's 1991 model. In this model, individuals' decisions to participate in some costly collective action depend on their private motivations to act and the level of visible participation of others in the population.

The model has three parameters: the size of the population, N, and the mean (b_{mean})

[19] Code in Java for this model is available at `http://people.duke.edu/~das76/Research/SiegelAJPSAnalysis.zip`.

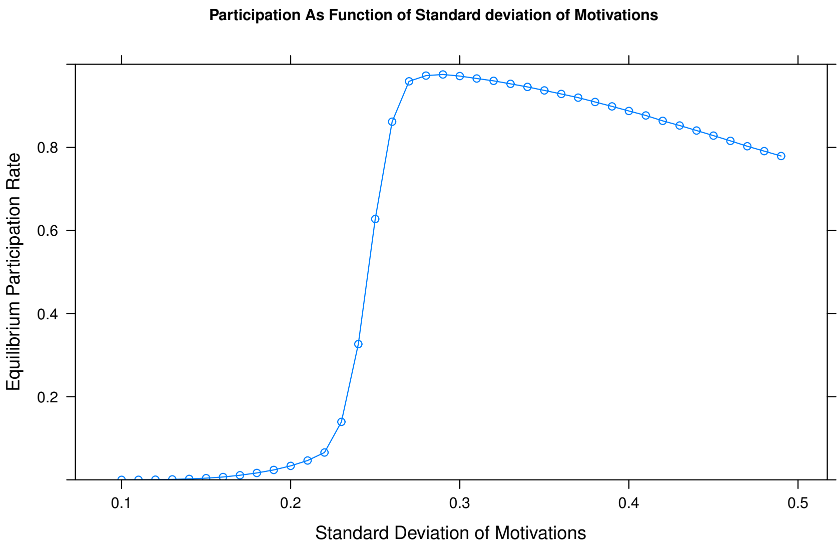

Figure 2: Analyzing M_0: Variation in b_{stdev}

and standard deviation (b_{stdev}) of the normal distribution representing the population's motivations to participate.[20]

Next, we analyze M_0. We have assumed known (and so fixed) values of N and b_{mean};[21] thus, analysis entails only computing comparative statics with respect to b_{stdev}. Figure 2 captures this analysis and allows us to see quickly that there are three different regions of interest: low, intermediate, and high. I posit that these are regions of common behavior, and label them as having "weak," "intermediate," and "strong" motivations, respectively. I will need to check this conjecture on more complex models as they will not be assumed separable, though it is trivially true in the one-parameter M_0.

I now propose a more complex model, M_1, that adds a one-time government intervention. I model this intervention as a total amount by which to reduce motivations, G, which gets divided equally across the population in the absence of bureaucracy. In Figure 3, I display the effect on participation of varying G (on the x-axes), conditional on values of

[20] As the model is not ergodic, the starting level of participation is also needed as a parameter. However, conditional on a set of initial conditions, the model evolves deterministically to an absorbing state, and we can describe the comparative statics of the model with respect to this state. I assume no one begins participating, as in Siegel (2009).

[21] Respectively 1000 and 0.6; see Siegel (2009) for justifications.

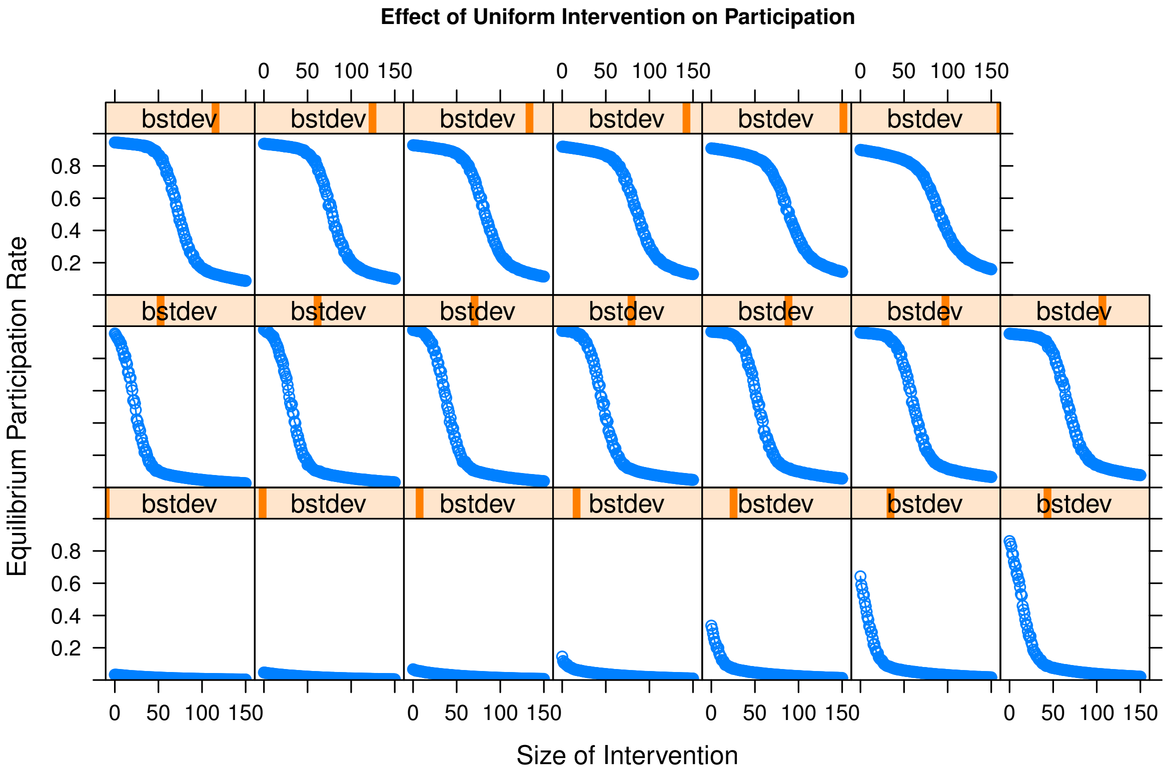

Figure 3: Analyzing M_1: Variation in G

b_{stdev} (represented by the lines in the bars above each conditional plot). I do this both as way of analysis of M_1 and as a check of my conjecture on the labels in M_0. Had I assumed an independent model addition in M_1, this check would not have been necessary. I could have instead analyzed variation in G only at representative parameters in each of the three labeled regions.

Figure 3 illustrates several things. One, my conjecture holds under the standard of "roughly the same shape" employed by SPS. Though participation depends on the interaction of G and b_{stdev}, violating perfect separability, within each of the three regions the shapes of the comparative statics with respect to G are roughly the same. Two, the shapes of the comparative statics vary by region, and suggest a different number of regions of common behavior in M_1 in each of the regions of M_0. When motivations are weak, there is only one region of common behavior in M_1, in which there is little effect of interventions. When they are intermediate, there are two regions: (i) small interventions are effective and increasing their level rapidly decreases participation; and (ii) past a cutoff, further increases in level have little effect. When motivations are strong, there are three regions: (i) small interventions have little effect; (ii) past a cutoff, increasing their level rapidly decreases participation; and (iii) past another cutoff, further increases have little effect. Taken together, there are six regions of the two-dimensional parameter space spanned by b_{stdev} and G to consider.

I make use of these regions to complete the model specification and analysis. The final complication is the addition of an information-gathering bureaucracy to form model $M_2 = M$. In line with the illustrative intent of the model, I capture this simply: given a bureaucracy of strength $S \in [0, 1]$, one receives a portion of the total intervention equal to $\frac{G}{N}((1 - S) + SR)$, where R is the ratio of that person's dissatisfaction to the average population dissatisfaction. Thus, more dissatisfied individuals receive more than an equal share, and stronger bureaucracies are better able to target the intervention.

In Figure 4, I display the effect of varying S. I consider representative parameter values that lie in five of the six identified regions of the parameter space, leaving aside the case of weak motivations in which little participation happens even without interventions. Additional analysis, not shown, indicates that varying b_{stdev} and G within these regions does not appreciably change the shape of the comparative statics, so SPS can be applied.

One aspect of the results in Figure 4 is as one would expect. In regions in which increasing intervention size has little effect on participation, bureaucracy also has little effect and is most likely not worth its cost. In regions in which interventions do have an effect, bureaucracy is increasingly effective the stronger it is. A second aspect is perhaps more subtle. In these latter regions, bureaucracy is more effective when motivations are strong than when they are weaker. The reason is that stronger bureaucracies are better at the very effective tactic of tamping down the motivations of very dissatisfied initial actors, and there are more of these types of individuals when motivations are strong than when they are weaker.

This completes our analysis. As in our trade example, we could further complicate this model while still using SPS as long we didn't induce strong dependence of comparative statics on precise values of model parameters from simpler models. So, for example, we could let the

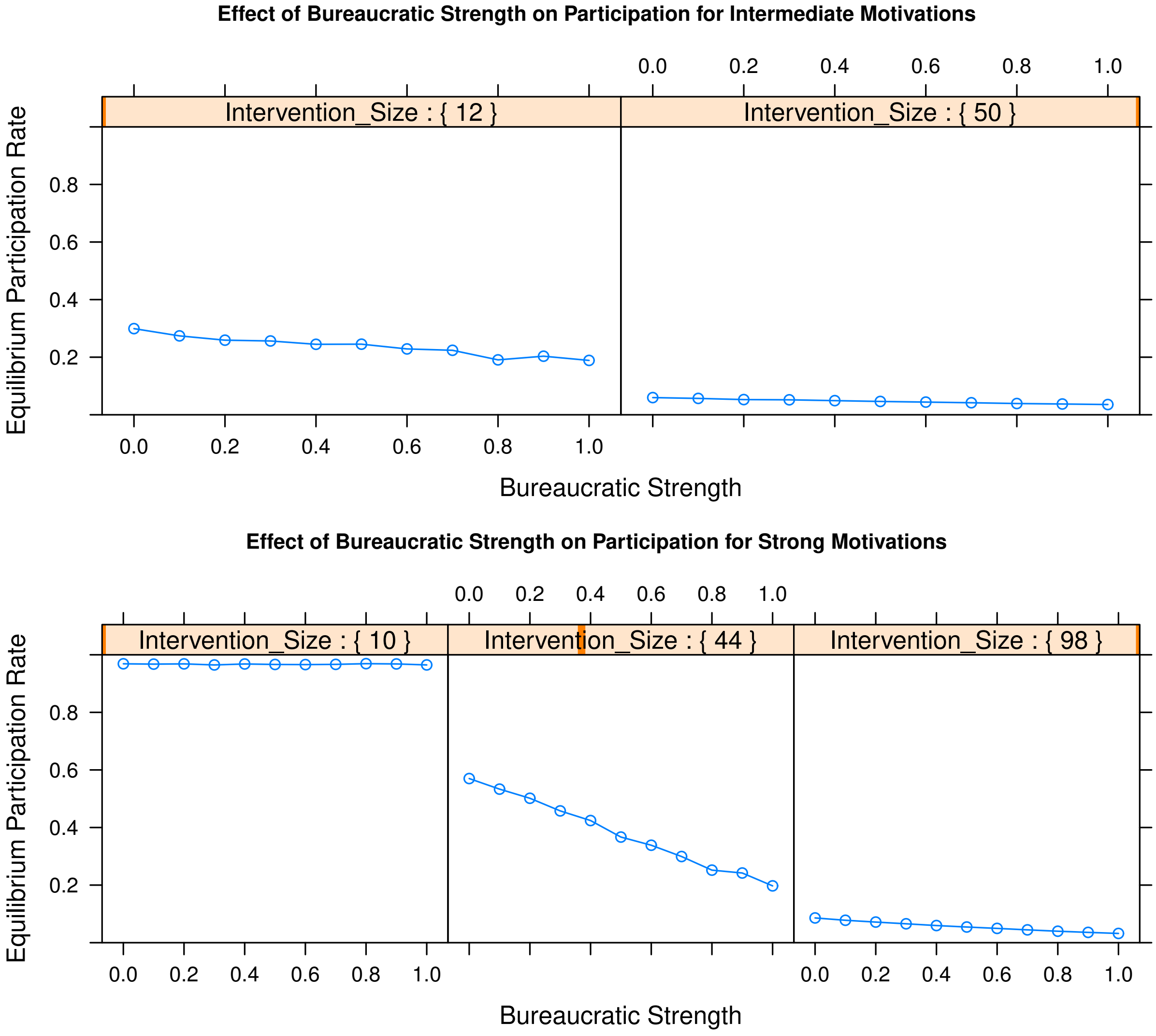

Figure 4: Analyzing M_2: Variation in S

strength of bureaucracy vary further according to any exogenous factor that didn't interact with existing model components, but we could not let it vary according to a factor that was strongly dependent on b_{stdev} and G. The latter scenario might occur in the context of bureaucrats who become more fearful of popular revolt as b_{stdev} increases, and more likely to take the money and run as G increases.

Conclusion

Computational modeling is a useful but underutilized approach for formally modeling complex political and social processes. I have addressed this underutilization in two ways. First, I proposed ways in which analysis of computational models could yield insights analogous to those derived from game-theoretic models. Second, I detailed a method, sequential parameter sweeping (SPS), to produce comparative statics in computational models that are readily comparable to those arising from game-theoretic analysis. One benefit of employing SPS is it provides for greater comparability between forms of modeling, hopefully leading to greater dialog between scholars. A second is that it can serve as a method for developing and analyzing computational models from the ground up, as following it—and stopping further model complexity when the method becomes no longer feasible—ensures that the model remains theoretically tractable despite the addition of further complexity.

A third benefit goes beyond the immediate utility of the method to computational modelers and game theorists in conversation with them. As empirical social science seeks to provide concrete answers to ever more complex problems, it becomes increasingly difficult to maintain the type of strong theoretical expectations that permit generalizable conclusions from the local identification of causal mechanisms. As illustrated by the program evaluation example provided above, computational modeling using SPS can be a useful tool for theoretical development in such situations. This point is accentuated by the fact that the computational tools necessary for modeling are often already possessed by empirical scholars. Using these tools to derive prior predictions of empirical outcomes can help both to structure and analyze experiments and to make arguments that empirical conclusions do not rest on unmeasured contextual factors, and therefore generalize to other contexts.

References

Scott Ashworth and Ethan Bueno de Mesquita. Monotone comparative statics for models of politics. *American Journal of Political Science*, 50(1):214–231, 2006.

Robert M Axelrod. *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press, 1997.

David P Baron and Michael C Herron. *A Dynamic Model of Multidimensional Collective Choice*, pages 13–48. MIT Press, 2003.

Jonathan Bendor, Sunil Kumar, and David A Siegel. Adaptively rational retrospective voting. *Journal of Theoretical Politics*, 22(1):26–63, 2010.

Jonathan Bendor, Daniel Diermeier, David A Siegel, and Michael M Ting. *A behavioral theory of elections*. Princeton University Press, 2011.

Scott De Marchi. *Computational and mathematical modeling in the social sciences*. Cambridge University Press, 2005.

Scott De Marchi and Scott E Page. Agent-based models. *Annual Review of Political Science*, 17:1–20, 2014.

Daniel Diermeier and Keith Krehbiel. Institutionalism as a methodology. *Journal of theoretical politics*, 15(2):123–144, 2003.

Tiberiu Dragu and Oliver Board. On judicial review in a separation of powers system. *Political Science Research and Methods*, 3(3):473–492, 2015.

Joshua M Epstein. *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2006.

James D Fearon and David D Laitin. Explaining interethnic cooperation. *American political science review*, 90(04):715–735, 1996.

Justin Fox and Matthew C Stephenson. Judicial review as a response to political posturing. *American Political Science Review*, 105(2):397–414, 2011.

Tim Groseclose. A model of candidate location when one candidate has a valence advantage. *American Journal of Political Science*, pages 862–886, 2001.

C Anthony Hunt, Glen EP Ropella, Sunwoo Park, and Jesse Engelberg. Dichotomies between computational and mathematical models. *Nature biotechnology*, 26(7):737–738, 2008.

Hyojoung Kim and Peter S Bearman. The structure and dynamics of movement participation. *American Sociological Review*, pages 70–93, 1997.

Ken Kollman, John H Miller, and Scott E Page. Adaptive parties in spatial elections. *American Political Science Review*, 86(04):929–937, 1992.

Timur Kuran. Now out of never - the element of surprise in the east european revolution of 1989. *World Politics*, 44(1):7–48, 1991.

Michael Laver. Policy and the dynamics of political competition. *American Political Science Review*, 99(02):263–281, 2005.

Michael Laver and Ernest Sergenti. *Party competition: An agent-based model*. Princeton University Press, 2011.

John H Miller and Scott E Page. *Complex adaptive systems: An introduction to computational models of social life*. Princeton University Press, 2007.

Scott E Page. Path dependence. *Quarterly Journal of Political Science*, 1(1):87–115, 2006.

David A Siegel. Social networks and collective action. *American Journal of Political Science*, 53(1):122–138, 2009.

Herbert A Simon and Albert Ando. Aggregation of variables in dynamic systems. *Econometrica: Journal of the Econometric Society*, pages 111–138, 1961.

Oleg Smirnov and James H Fowler. Policy-motivated parties in dynamic political competition. *Journal of Theoretical Politics*, 19(1):9–31, 2007.

Appendix for "Analyzing Computational Models"

David A Siegel

Sequential Parameter Sweeping More Formally

I provided an informal description of sequential parameter sweeping (SPS) in the text, making use of two examples to illustrate the method. In this appendix I formalize, to the extent possible, some of its assumptions and arguments. This formalization is useful in applying SPS to more complex models.

I begin by defining one's ideal model, M. This model contains all complexity the modeler believes is necessary to properly instantiate her theory. Since our focus is on the comparative statics of the steady states of computational models, I formally define M as a mapping taking a set of parameters, B, to a set of steady state variables, X. Each individual parameter is β_k, and each individual steady state variable is x_l.

Next I define a sequence of nested models indexed by i, M_i, with $M_m = M$ for some finite m. These have a very specific property: each M_i is nested within all M_j with $j > i$. Formally, to be nested means, for all $j > i$: (i) the set of parameters, B_i, in each model M_i is a strict subset of the set of parameters, B_j, in each M_j; and (ii) the set of steady-state variables, X_i, in each model M_i is a (not necessarily strict) subset of the set of steady state variables, X_j, in each M_j. Practically, this means that one forms nested models by removing one or more parameters and possibly one or more steady-state variables from a more complex model. This nesting is not typically unique, and should be informed by the substantive question. In the interpersonal trade example in the text, the PD addressed a substantively distinct question on its own, and so made sense as a nested component of the more complex model. In the program evaluation example in the text, collective action absent government is one question, the effect of government interventions on collective action is a second, and the effect of bureaucracy on the effect of interventions is a third.

Finally, I define a separability property that connects the analyses of the M_i. Let B_{in}, $n = 1 \dots N_i$, N_i finite for all i, be disjoint regions of the parameter space of model M_i, such that the union of the B_{in} is the parameter space spanned by the set of parameters B_i. In other words, the B_{in} partition the parameter space of a model M_i. We say the parameter space of model M_i is *separable into regions of common behavior* if three conditions are met.

The first condition is that within each region B_{in} the functional dependence of each of the steady state variables in X_i on each of the parameters in B_i must "look the same." Here "look the same" refers only to the *shape* of the comparative statics within each region, and not to the boundaries between regions or the precise values of each endogenous variable within regions. That is to say, what is important is that the behavior of the model M_i within each region, as represented by the manner in which the endogenous variables change with the exogenous parameters, is similar within the region. One must care about both sign and rate of change of the comparative statics in assessing this condition, and extant theory is particularly helpful in justifying adherence to it. The two examples in the text illustrate what it means to satisfy this condition.

The first condition is, to some extent, deliberately vague: the intent is to capture different regions of the parameter space within which the causal mechanisms driving the dependence of each steady state variable on its parameters are common to the region. As this deliberate vagueness comes up at several points in what follows, it is worth briefly elaborating on its intent before moving on to the second condition.

The intent of the first condition, and SPS in general, is to understand patterns of behavior, rather than point predictions. This same intent underlies comparative statics of game-theoretic models, and SPS is similar to the spirit of one method of deriving them, monotone comparative statics.[1]

To pursue this intent, we must define a standard for what constitutes a pattern of behavior. One is typically only concerned with the signs of comparative statics in game theory models. A pattern of behavior according to this standard is a specification, possibly condi-

[1] Thanks to Ken Shotts for the analogy.

tional, of the signs of all comparative statics across all subsets of the full parameter space.

SPS uses a somewhat different standard: it requires that not only the sign, but also the rough shape of the comparative statics must be similar within each region of common behavior. This standard is stricter than that typically used in game theory in the sense that it is not just the sign that matters, but it is weaker in that the best one can do in a computational model, particularly when models exhibit substantial stochasticity, is to specify a numerical tolerance for what constitutes rough similarity.

The phrase"look the same" arises from this consideration of numerical tolerance. I could have replaced this with a more formal definition along the lines of "the integrated difference of the comparative statics taken at two different parameter values within a region of common behavior must not exceed y across the domain of the region" or "the sign of the comparative statics should not cross more than y beyond zero from above to be considered positive." However, this would have reduced flexibility too much for the end-user of the method, who should remain free to specify whatever standard of "looking the same" is justified by the research context in question.

An end-user of the method could vary the standard of numerical tolerance embedded in SPS in more substantive ways if desired. In some cases, the analyst may find only the sign of the comparative statics of interest, and possess broad tolerance as to numerical variability near zero. In others, the analyst may want a more precise depiction of the complete functional form of the comparative statics, with relatively little error. The former might occur when one's model is intended to be suggestive of an underlying behavior, while the latter might occur when one believes one's model has substantial ground truth, perhaps because many parameters have been fitted to data. Somewhere between the two is the standard I have used in this presentation, which was chosen under the presumption that the shapes of the comparative statics capture the most important patterns of behavior to the researcher.

However, when considering changing the standards of SPS more substantially, it is important to note that the standard used does affect the degree to which one can employ SPS successfully. A stricter standard of "looking the same" will generally imply less ability to discover regions of common behavior, and SPS will generally fail more quickly under this stricter standard. In contrast, a very weak standard that merely focuses on the sign of the comparative statics rather than their shape would fail less often than the one I have presented.

I now return to discussing the separability property. To specify its second condition, we need to link nested models. Recall that the shapes of the comparative statics, rather than the precise parameter boundaries that specify each B_{in}, are what truly define each region. Therefore, define regions B'_{jn} of the parameter space of model M_j, $j > i$, in the following manner: In each B'_{jn}, the comparative statics for each parameter in B_i, for every steady state variable in X_i, share roughly the same functional forms as they do in the corresponding region B_{in} of the parameter space of the simpler model M_i. In other words, the comparative statics whose similar behavior defined the B_{in} in model M_i also define the B'_{jn} in model M_j for any $j > i$. Note that because the more complex model M_j has a higher-dimensional parameter space than M_i, the B'_{jn} need not be unique, since they are specified only by the similar behavior of comparative statics with respect to parameters and steady state variables present in the simpler model M_i.

The second condition is that regions B'_{jn} exist for each model M_j, for all $j > i$, such that within each region B'_{jn} the functional dependence of each of the steady state variables in X_i on each of the parameters in B_i must "look the same," as in the first condition. Put another way, our first two conditions on separability of a model M_i imply that as we build up more complex models from M_i, we must still be able to find regions of the parameter space for every model in which all comparative statics that were present in M_i continue to betray roughly the same functional dependence within each region as they did in M_i.

The third and final condition is the most stringent, as it specifies a particular kind of independence between models. It states that within each of the B'_{jn} that satisfy the second condition, for each model M_j, for all $j > i$, the functional dependence of each of the steady state variables in X_j on each of the parameters in B_j that are not also in B_i must be approximately independent of the parameters in B_i. As in the first two conditions, this is left deliberately vague, for the same reasons as above. That said, approximately independent admits a relatively easy empirical test: if one's comparative statics for parameters that only exist in model M_j vary substantially with parameters that exist also in M_i for $j > i$, beyond variation due to being in different regions B'_{jn}, then this condition is violated. A cross-partial derivative of an endogenous variable with respect to a parameter in B_i and a parameter that is in B_j but not B_i that is significantly different from zero at a point within the interior of a region B'_{jn} would indicate a violation of this condition.[2] In other words, the comparative statics with respect to the new parameters added in any more complex model cannot depend significantly on the parameters in the simpler model M_i, beyond any dependence captured by presence within a region B'_{jn} of the parameter space of M_j.

Put slightly differently, the third condition specifies that the effects of all parameters that are in more complex models M_j, $j > i$, but not in the simpler model, M_i, are approximately independent of all parameters in M_i, apart from the role of M_i's parameters in determining the boundaries of the regions of the parameter space. This way of phrasing connects separability to labeling. A nested series of models possesses separability if one can find regions of the parameter space of each M_i, $i < m$, that can be labeled; additional parameters in M_j, $j > i$, but not in M_i can have label-dependent effects on steady-state variables, but not significant effects dependent on the particular values of the parameters in M_i beyond this. Identification of these regions can require both computational characterization of M_i and extant theory.

With separability defined, we can move to specifying SPS. SPS begins by instantiating a simplified version of the model, M_0, with no more than 3 parameters. The reduced number of parameters in M_0 should allow one to completely specify comparative statics for each model parameter in all equilibria/steady states/limiting distributions of M_0. This can be done by direct computation. SPS then iterates the three steps below in order to instantiate a particular sequence of models, M_i, that increase in complexity. This iteration continues until

[2] Leaving open what counts as "significantly different" from zero maintains the needed flexibility discussed above.

either: (i) $M_i = M$ for some i, implying the full model is analyzable with SPS; or (ii) step one fails for some i, implying only part of the model is analyzable with SPS. In the latter case one could either: stop at the most complex model analyzable by SPS; analyze the full model using a sampling procedure, supplementing insights from this analysis with those arising from the partial SPS approach; use a non-nested modeling approach that implicitly ignores interactions between model components; or reformulate one's model in a more hierarchical fashion so as to make more effective use of SPS. The steps to be iterated are as follows, with $i = 0$ in the first pass and M_0 assumed fully analyzed.

1. If possible, separate the parameter space of model M_i into regions of common behavior (see definition above).[3]

2. If step one yields no such regions, or an unmanageably large number of them, cease SPS. Otherwise, iterate model M_i toward model M. Call this new, more complex model M_{i+1}; it should possess no more than 1-2 additional parameters. This iteration should be chosen to make step three possible.[4]

3. Fully characterize all possible equilibria/steady states/limiting distributions of this more complex M_{i+1} and compute comparative statics for each of its new variables and parameters (i.e., variables and parameters that are in M_{i+1} but not M_i). Comparative statics must be computed within each of the regions identified in step one, at representative values of the parameters in M_i that distinguish these regions. In other words, for every labeled region from step one, one must choose a set of representative values of the parameters in M_i that delineates that region; these representative parameters

[3] Multiple such partitions may exist. These can be distinguished on the basis of substantive considerations (e.g., which partition has the clearest substantive meaning), or on which allows SPS to proceed the furthest.

[4] Ideally the order of adding complexity would not matter. However, should separability fail before model M is reached, one must decide which aspect of complexity to add first. Again, this decision can be made on the basis of substantive considerations (e.g., which is the most substantively interesting aspect of complexity), or on which order allows SPS to proceed the furthest. This decision can also be influenced by what approach one desires to take with respect to analyzing M. For instance, if one plans to analyze a series of non-nested models due to the inability of SPS to fully analyze M, then the order should be chosen to enable these non-nested models to tell the clearest story.

are then used to compute comparative statics for the new parameters added in step two. If $M_{i+1} = M$ this step completes SPS; if not, let $i = i + 1$ (i.e., increment the index i by 1) and return to step one.