Exploring the Dynamics of Variational Inequality Games with Non-Concave Utilities

Ian Gemp
College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01003
imgemp@cs.umass.edu

Abstract

Variational inequality (VI) theory has proven useful in modeling and analyzing a variety economic markets. However, in order to ensure the analysis is tractable, models are usually constrained to an unrealistic regime of concave utilities and monotone operators undermining the reliability of real-world conclusions such as the uniqueness and location of equilibria. We argue that machine learning can help address this issue. In this paper, we ignore typical monotonicity requirements and construct a generic, yet more realistic market model possessing several desirable qualities. We then borrow a tool from dynamical systems to cope with our model's lack of theoretical guarantees. Additionally, in order to handle the large size of standard VI game formulations, we further enhance the tool to accommodate more sophisticated numerical algorithms and propose a heuristic for efficient use of generated trajectories. We illustrate these enhancements by applying the resulting pipeline in the context of cloud services.

1 Introduction

Variational inequalities (VIs) extend fixed point problems to the *constrained* domain enabling the study of many interesting subjects including constrained mechanics (*rigid boundary*), game theory (*strategy simplex*), reinforcement learning (*function approximation*), traffic flow (*non-negative flows*) and finance & economics (*non-negative prices/quantities*) [9, 8, 3, 5, 11]. The following definition characterizes solutions to VI(F, K).

Definition 1. *Variational Inequality Problem VI(F, K):*
Find $x^* \in K \subset \mathbb{R}^n$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0, \; \forall x \in K$$

where $F : K \rightarrow \mathbb{R}^n$ is a given continuous function, K is a given closed convex set, and $\langle \cdot, \cdot \rangle$ is the standard inner product in $\mathbb{R}^n$.

The simplest algorithm for solving VIs is the (Euclidean) projection algorithm, $x_{k+1} \leftarrow P_K(x_k - \alpha F(x_k))$, otherwise recognized as projected gradient descent in the optimization literature. Hence, x^* is also a solution to the VI when $x^* = P_K(x^* - F(x^*))$ making clear its extension to unconstrained fixed point problems.

The basis for many theoretical results of VIs applied to game theory is built on the convexity of loss functions and, more generally, the monotonicity of F.

Definition 2. *Differentiable function f is convex (strictly, strongly with $c > 0$) iff $\forall\, x, y \in K,\; x \neq y$*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0 \;\; (> 0, \geq c||x - y||^2)$$

and pseudo-convex (strictly, strongly with $c > 0$) iff

$$\langle \nabla f(x), y - x \rangle \geq 0 \implies \langle \nabla f(y), y - x \rangle \geq 0 \ (> 0, \geq c||x - y||^2)$$

Replace ∇f with F for (pseudo)-monotone; (pseudo)-convex $f \implies$ (pseudo)-monotone ∇f.

A restricted class of VIs is equivalent to a Nash equilibrium (NE) problem [4]. Let $L_i(x_i, x_{-i})$ be a pseudo-convex loss function for player $i \in \{1, \dots, n\}$ defined for x_i in convex set K_i. Let $F = (\nabla_{x_1} L_1, \dots, \nabla_{x_n} L_n)$ and $K = \prod K_i$, then the solutions, x^*, to VI(F, K) are also solutions to the NE problem: find $x^* \ s.t. \ L_i(x_i^*, x_{-i}^*) \leq L_i(x_i, x_{-i}^*) \ \forall i, (x_i, x_{-i}^*) \in K$.

Although this equivalence is only restricted to pseudo-convex losses, loss functions are nearly always restricted to be strongly convex [19, 12, 14]. It is well known that the VI arising from strongly convex losses is strongly *monotone* and admits a unique, globally exponentially stable solution, which simplifies the analysis and eases algorithmic convergence [13].

Unfortunately, VIs arising from pseudo-convex losses do not enjoy the same guarantees. In parallel with convex analysis, the sum of monotone operators (convex functions) is monotone (convex), however, the sum of pseudo-monotone operators (pseudo-convex functions) is not necessarily pseudo-monotone (pseudo-convex) and so not much can be said about the resulting VI [7]. Thus, there is this barrier to model expression. While the necessary progress in VI theory may not emerge for some time, machine learning approaches can alleviate some of the trepidation experienced when analyzing more complex models. In the rest of the paper, (Section 2) we explain how VI's connection to projected dynamical systems allows us to apply a Monte-Carlo sampling tool for analyzing complex VI models; (Section 4) we enhance this tool so it scales to large games (i.e. many players); (Section 3) we discuss an interesting application in modeling the cloud services market economy and survey previous modeling attempts; (Section 6) we explore our proposed model with a hypothetical case study and demonstrate the proposed machine learning pipeline on our new cloud services model.

2 Identifying Boundaries of Attraction

As discussed above, VI theory provides no general guarantees on the uniqueness of Nash equilibria when losses are pseudo-convex. This motivates an *algorithmic* approach to identifying the number of equilibria, their locations, and possibly other phenomena. In particular, we will leverage theory and algorithms from dynamical systems - we refer the interested reader to [20] for a gentle introduction.

[13] established an equivalence between VIs and projected dynamical systems that brings new theory and algorithms, providing a foundation for the necessary analysis.

Definition 3. *Assuming that the feasible set K is a convex polytope, the projected dynamical system, PDS(F, K), corresponding to VI(F, K) is $\dot{X} = \Pi_K(X, -F(X))$ with $X(0) = X^0$ and $\Pi_K(X, -F(X)) = \lim_{\delta \to 0} \frac{P_K(X - \delta F(X)) - X}{\delta}$.*

In terms of attractors, strongly monotone VIs admit only stable fixed points accompanied by a relatively small range of attractor dynamics including stable spirals and nodes. As expected, less can be said of VIs arising from pseudo-convex loss functions. Other, qualitatively distinct attractors include limit cycles, tori, and strange attractors (see Figure 1). It's important to be aware of these other possible attractors when analyzing a more complex system. For example, a stock opening in one range of prices may cause the group of stocks as a whole to simply *readjust* to a new stable NE. On the other hand, opening the stock in another range of prices may result in the group tending towards a limit cycle where prices continuously oscillate. It's then obvious that the ability to predict which ranges result in which behaviors helps determine where it's best to open the stock. Thus, we would like to identify the endpoints of these ranges, or more generally, the *boundaries of attraction* (BoAs).

There are several existing techniques for identifying BoAs. The theory of Lyapunov functions has long motivated a large group of these, however, they can only be applied to restricted types of nonlinear systems and are not capable of identifying the entire BoA [10]. Others attempt to approximate Lypapunov functions using a set of scalar functions [15]. Still other, non-Lypunov based approaches have been proposed that work backwards from the attractor. These methods tend to be lightweight, but less reliable. Recently, [2] developed a method for identifying BoAs that relies on Lyapunov exponent (LE) theory. Convergence of LEs can be slow, but they enjoy the advantage of being independent of initial conditions and can be applied to general nonlinear systems. The authors proposed the use of Monte-Carlo sampling to alleviate the computational load of calculating LEs. Their approach can give us an idea of the number and types of attractors we can encounter in a bounded space, but first, to understand their algorithm, we need an understanding of LEs.

LEs measure the long-term deformation of a sphere along a trajectory in the dynamical system and are invariant within a single BoA. It's this invariance property that allows us to use the LE as a signature for the basin of attraction in spite of varying initial conditions.[1] Furthermore, LEs reveal the type of attractor. For instance, if all values in the LE are negative, the attractor is a stable fixed point; if instead, one of the values is zero, the attractor is a limit cycle.

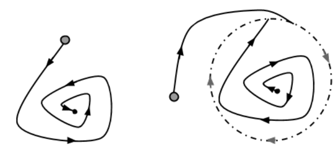


Figure 1: Stable spiral (left) and limit cycle (right, dashed)

The general idea of [2]'s algorithm is to sample grid points with high probability of being near a BoA, compute the LEs of the sampled grid point as well as a few of its neighbors, and then compare LEs between all pairs of tested points. If a pair of LEs do not match, then they are located on either side of a BoA and the pair can be added to a training set for a classifier (e.g. SVM). In addition, the probabilities of the neighbors can be increased since they are most likely near the boundary as well. In the case where the LEs are the same (within some tolerance), the probabilities can be reduced.

In their paper, they consider domains in $\mathbb{R}^2$ to $\mathbb{R}^4$. Low dimensionality allows them to apply standard LE calculation techniques coupled with more basic ODE solvers (e.g. constant step size) without compromising runtime. We are more interested in the high dimensional domains that often occur in VIs with many players, each of which controls multiple variables. Given a constant number of grid points per dimension, the total number of grid points scales exponentially with the number of dimensions and quickly makes this Monte-Carlo sampling approach impractical. Moreover, basic ODE solvers may incorrectly track the trajectories of systems that contain multiple time scales.

3 Cloud Services Market

In line with our statements above, previous attempts at analyzing the cloud services market assume a model with convex (concave) loss (utility) functions. [1] and [6] both separately formulate a single-cloud, multi-client market as a Stackelberg game with strongly-concave, differentiable client utilities where the cloud adjusts prices and the clients adjust their service demands. They define quality of service (QofS) as a function of the cloud congestion, an idea borrowed from the study of radio and mobile networks. Neither considers cost functions associated with scaling cloud services.

[21] design a multi-cloud, multi-client system with agents (brokers) that sell cloud services to clients at a commission. Their focus is on system design with real world applications rather than equilibrium analysis, however, their work provides a useful blueprint for model construction. They argue that congestion-based QofS is unrealistic and client demand functions must be monotonically decreasing with a nonzero critical point after which utility remains zero. More importantly, they define their client utilities to be quasi-concave, piece-wise linear suggesting that simple concave utilities are oversimplifications for the cloud services market.

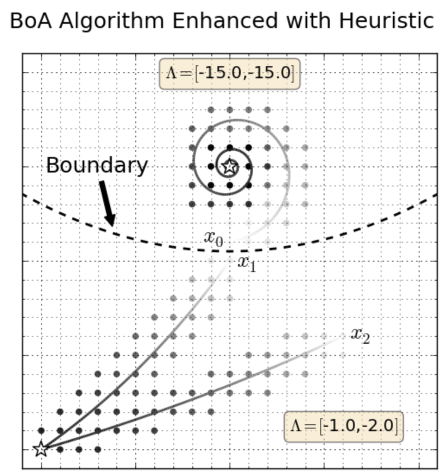
4 Improving the BoA Identification Algorithm

As stated, we would like to alter the BoA algorithm so it scales more gracefully with dimensionality. The first step is to adjust the LE computation to be able to accompany an ODE solver ($\mathbb{S}$) with an adaptive step size scheme ($\mathbb{T}$). While the fix is somewhat trivial, it was very difficult to come across explicit LE computation instructions for constant step sizes [22, 18] and we never found any such instructions for adaptive step sizes. We include the necessary pseudocode in Algorithm 1.

Next, we point out that computing an LE involves following the trajectory from an initial point x^0 until convergence. The runtime for this computation alone can be extensive for high dimensional systems. Since the LE is a global property and hence, in theory, a property shared by all points along the trajectory, ignoring the computed LE's association with all points along the trajectory seems

[1] Two basins may have the same LE though.

particularly wasteful. Instead of throwing out this information, we can include it by recognizing that all subsequent points after the initial point along the trajectory are ideally progressing away from the boundary (assuming integer dimensional BoA's). Moreover, the LE gives us an idea of the exponential rate of *divergence* away from the boundary, and so we can use the LE to decay the probability of grid points along the trajectory. Algorithm 2 describes the steps used to adjust probabilities using this heuristic and an example is displayed in Figure 2. This approach allows us to update the probabilities of many more grid points per LE computation, helping to combat the issues of dimensionality.



Algorithm 2 Update Grid Probability Along Trajectory $\boldsymbol{x}$

INPUT: LE, $\boldsymbol{x}$, Δt^k, d_{max}

Initialize hashes N, D
$t = 0, T = \sum \Delta t^k, \lambda = \max(|\text{LE}|)$
for x^k in $\boldsymbol{x}$ **do**
  $\boldsymbol{g}, \boldsymbol{d}$ = gridNeighbors&Distances(x^k)
  for each (g, d) in $(\boldsymbol{g}, \boldsymbol{d})$ **do**
    $N[g] \stackrel{+}{=} e^{-\lambda \cdot t/T} \cdot \Delta t^k$
    $D[g] \stackrel{+}{=} \Delta t^k$
  end for
  $t \stackrel{+}{=} \Delta t^k$
end for
for each g in N, D **do**
  $P(g) \stackrel{*}{=} N[g]/D[g]$
end for

Figure 2: The probabilities of points farther along the trajectory (white to black) should be reduced as they are most likely far away from any boundary. These adjustments can be shared with the surrounding grid points.

5 A New Market Model

Our aim is to formulate a multi-cloud, multi-client model that trades strong concavity for qualitative realism and a congestion-based QofS for an explicit quality control variable with cost impacts. In particular, our goal is to model the prominent, commercial cloud market that has arisen over the past decade that invites clients both small and large. For this reason, we model the market with uniform client prices and service degradation (1/quality) meaning each cloud i advertises the same price-degradation pair, (p_i, d_i), to every client j. As suggested by [21], client j's demand for cloud i, Q_{ij}, is monotonically decreasing in p_i and d_i with a nonzero *zero-utility* cutoff. Note that while we will continue to discuss this model in the context of cloud services, our model can likely be applied to any industry where firms set prices for QofS at a cost to themselves.

Whereas in previous work, a **single** cloud price level is set by an optimal response function and **many** clients are modeled by utility functions, our formulation instead incorporates the **multiple**

Algorithm 1 LE for Adaptive Step Sizes

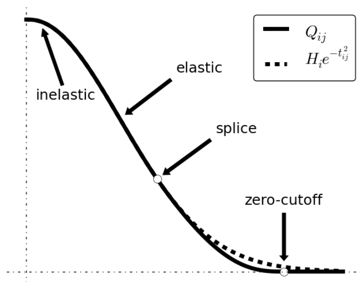
INPUT: $F, K, x^0, \Delta t^0, \mathbb{S}, \mathbb{T}$

$\Lambda = (0, \dots, 0), \psi^0 = \mathbb{I}, k = 0, T = 0$
J $\leftarrow$ Jacobian(F(x)) $\cdot\ \psi$
GS $\leftarrow$ GramSchmidt **without** normalization
$|\cdot|_c \leftarrow$ column-wise normalization
repeat
  $x^{k+1} = \mathbb{S}(x^k, \Delta t^k, -F)$
  $\hat{\psi}^{k+1} = \mathbb{S}(\psi^k, \Delta t^k, -J)$
  $\hat{\psi}^{k+1} = \text{GS}(\hat{\psi}^{k+1})$
  $\lambda \Delta t = \log(|\hat{\psi}^{k+1}|_c)$
  $\Lambda = (\Lambda \cdot T + \lambda \Delta t)/(T + \Delta t^k)$
  $T = T + \Delta t^k$
  $\psi^{k+1} = \hat{\psi}^{k+1}/|\hat{\psi}^{k+1}|_c$
  $\Delta t^{k+1} = \mathbb{T}(x^k, x^{k+1}, \psi^k, \psi^{k+1}, \Delta t^k)$
until Convergence of Λ

client responses through demand functions and the **many** cloud interactions through utilities (profit functions π_i); in other words, the roles are swapped.

We desire three additional qualities of our demand function. First, when price/degradation falls below some threshold, client demand should be relatively indifferent (inelastic). For example, client demand may not increase if prices drop because they've already reserved enough compute space to satisfy all their needs. Similarly, client demand may not increase if the cloud cuts delay time from 2 nano-seconds to 1 nano-second because in the grand scheme of things, computation already appears instantaneous. Second, there exists a region of prices/degradation where demand is elastic; otherwise, clients would be indifferent to any change in cloud services making this a very boring problem. Third, the demand functions should be continuous and differentiable to match Definition 1. We actually enforce twice-differentiability because our LE computation relies on the second derivative information (Jacobian of $F(x)$), however, this isn't a strict requirement.

Many common demand functions satisfy the second requirement such as the linear, quadratic, and negative-exponential demand functions, however none incorporate all three requirements. We define the function below to fill this gap. Our demand function, Q_{ij}, consists of a squared-exponential spliced with a 5th degree polynomial (coefficients β in Appendix). The function is twice differentiable, contains both elastic and inelastic regions, and drops to zero-demand at finite t_{ij} (see Figure 3). The cosine function is also a viable candidate, however, there exists an approximate linear transformation to the squared exponential [17], so our choice simply reflects personal preference. We've also included factors $p_r = \frac{p_i}{\bar{p}}, d_r = \frac{d_i}{\bar{d}}$ where $\bar{p}$ and $\bar{d}$ are cloud price and degradation averages so that clients are also attracted to low prices/degradation in a relative sense. Client-cloud loyalty is simulated through client j's elasticity coefficient, α_{ij}, while purchasing power is given by H_{ij} (see equations 1 & 2).



$$t_{ij} = \alpha_{ij} p_i d_i p_r d_r \tag{1}$$

$$Q_{ij} = \begin{cases} H_{ij} e^{-t_{ij}^2} & , t_{ij} \in [0, t^c] \\ \sum_{k=0}^{5} \beta_k t_{ij}^i & , t_{ij} \in (t^c, t^c + 1) \\ 0 & , t_{ij} \in [t^c + 1, \infty) \end{cases} \tag{2}$$

$$\pi_i = \sum_j \underbrace{p_i Q_{ij}(p_i, d_i)}_{revenue} - \underbrace{\frac{c_i}{d_i^2} Q_{ij}(p_i, d_i)}_{cost} \tag{3}$$

Figure 3: Proposed demand function $Q_{ij}(t_{ij})$, $t^c = 1$.

Cloud profit[2], π_i, is defined as revenue minus cost where cost scales as the square of quality ($1/d_i$) with coefficient c_i.

Let $x_i = (p_i, d_i) \in [\epsilon, \infty)^2$, $i \in 1, \ldots, n$, and $L_i(x_i, x_{-i}) = -\pi_i$, then we would like to analyze the model given by VI(F, K) where $F = (\nabla_{x_1} L_1, \ldots, \nabla_{x_n} L_n)$ and $K = [\epsilon, \infty)^{2n}$. Note that K is unbounded (not compact), so we are not guaranteed a solution to the VI exists.

We stated in the introduction, an equivalence between the VI with pseudo-convex losses and the NE problem. The cloud profit functions, as defined, are, in general, non-concave. Although we no longer have a guarantee that solutions to the VI are necessarily Nash equilibria, we still have an equivalence between VI(F, K) and PDS(F, K). This means we can perform the same BoA analysis, but we'll need to check stable fixed points to see if they satisfy the Nash definition, which amounts to solving n non-convex, 2-d, constrained optimization problems. In our solution, we use Scikit-learn's *L-BFGS-B* for this task [16]; runtime is negligible relative to the BoA algorithm.

[2] π_i is nondifferentiable at $d_i = 0$, however, zero price and infinite quality are nonsensical, so our market is constrained to $[\epsilon, \infty)$

6 Experiment

To demonstrate the promise of the described pipeline, we focus on identifying the BoA's (as well as Nash equilibria) of our proposed cloud services market economy model. Here we investigate a hypothetical scenario in which four cloud companies compete for the opportunity to provide service to five clients looking to transfer their in-house computation to the cloud. The first three cloud companies are large providers with highly optimized servicing capabilities (lower c_i), while the last two are newcomers to the market, trying to fill a niche with higher cost green-tech (higher c_i). Client 1 is a big buyer loyal to clouds with the 3 lowest cost functions (e.g. big name providers). Client 2 is a medium buyer with slight preference towards green-tech. Client 3 is a small buyer who prefers green-tech, but is not opposed to a large corporation. Client 4 is a big buyer loyal to cloud 1, but otherwise prefers green-tech. To compute LEs, we're using a projected version of Heun-Euler, a 2nd order, explicit ODE solver with an adaptive step size.

Running the BoA algorithm[3] over a 10 dimensional grid (6 points/dimension) with the enhancements described in Section 4 returns a set of of positive-negative samples for each reference LE. After running an SVM on each LE sample set, we define the boundaries as the critical points at which the SVM with the highest margin prediction is dethroned by an SVM with a higher margin prediction.

In Figure 4, we consider a scenario where green-tech newcomer, cloud 5, enters the pre-established cloud services market described above. Opening with (p_5, d_5) in either of the stable regions sets the market on a path toward the same NE; the two regions are mislabeled as distinct due to noise in their LE calculations. On the other hand, launching their business in the unstable region results in chaos and should be avoided. Although we can't visualize both green-tech newcomers entering the market (>3d), we can quickly evaluate our SVM classifiers to determine the corresponding basin of attraction and associated characteristic LE for any given set of price-degradation pairs. Obviously, there are factors that our model does not take into account. In spite of this, knowledge of BoAs combined with market monitoring can also be used to suggest when a discussion of external intervention might be prudent (e.g. government regulation) or when external intervention might transition the market into a more desirable basin of attraction.

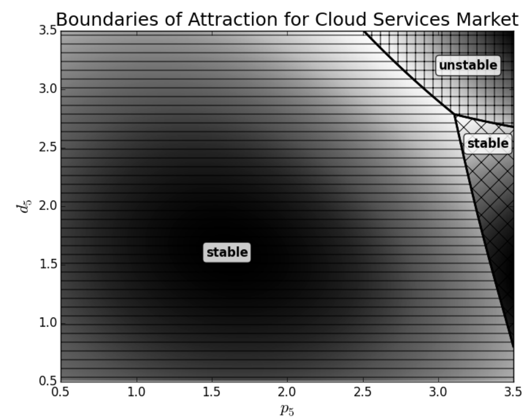


Figure 4: Basins of attraction are marked stable or unstable and differentiated by pattern, each with a gradient that runs from most likely belonging to the region (dark) to least likely (light). Boundaries are marked by black lines.

7 Conclusion

In this paper, we extend a dynamical systems BoA identification algorithm to serve as a tool for analyzing previously untenable variational inequality markets. We then apply this tool to a novel model of the cloud services market economy that incorporates a number of qualitatively realistic features found lacking in related work. Finally, our demand function is general enough to be reused in the analysis of a variety of economic markets.

In future work, we would like to derive a more principled algorithm for identifying the boundaries of attraction. We are currently considering a Bayesian approach using gaussian processes and a tailored acquisition function. This sort of approach is very appealing as it removes the need for a grid and allows for noisy observations of the LEs.

[3] All code at https://github.com/all-umass/VI-Solver

References

[1] Ashraf Al Daoud, Sachin Agarwal, and Tansu Alpcan. Brief announcement: Cloud computing games: Pricing services of large data centers. In Idit Keidar, editor, *Distributed Computing, 23rd International Symposium, DISC 2009, Elche, Spain, September 23-25, 2009. Proceedings*, volume 5805 of *Lecture Notes in Computer Science*, pages 309–310. Springer, 2009.

[2] Ali Reza Armiyoon and Christine Qiong Wu. An innovative approach for identifying boundaries of a basin of attraction for a dynamical system using Monte Carlo techniques and Lyapunov exponents. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pages 6299–6304. IEEE, 2014.

[3] Dimitri P. Bertsekas. Temporal difference methods for general projected equations. *IEEE Transactions on Automatic Control*, 56(9):2128–2139, Sep 2011.

[4] E. Cavazzuti, M. Pappalardo, and M. Passacantando. Nash equilibria, variational inequalities, and dynamical systems. *Journal of Optimization Theory and Applications*, 114(3):491–506, 2002.

[5] S. Dafermos. Traffic equilibria and variational inequalities. *Transportation Science*, 14:42–54, 1980.

[6] Makhlouf Hadji, Wajdi Louati, and Djamal Zeghlache. Constrained pricing for cloud resource allocation. In *Proceedings of The Tenth IEEE International Symposium on Networking Computing and Applications, NCA 2011, August 25-27, 2011, Cambridge, Massachusetts, USA*, pages 359–365. IEEE Computer Society, 2011.

[7] N. Hadjisavvas, S. Schaible, and N.-C. Wong. Pseudomonotone operators: A survey of the theory and its applications. *Journal of Optimization Theory and Applications*, 152(1):1–20, 2012.

[8] Patrick T. Harker. Generalized Nash games and quasi-variational inequalities. *European Journal of Operational Research*, 54(1):81–94, Sep 1991.

[9] P. Hartman and G. Stampacchia. On some nonlinear elliptic differential functional equations. *Acta Mathematica*, 115:271–310, 1966.

[10] Alexanckr Levin. An analytical method of estimating the domain of attraction for polynomial differential equations. *Automatic Control, IEEE Transactions on*, 39(12):2471–2475, 1994.

[11] A Nagurney. *Network Economics: A Variational Inequality Approach, second and revised edition*. Kluwer Academic Press, 1999.

[12] A Nagurney and T Wolf. A Cournot–Nash–Bertrand game theory model of a service-oriented internet with price and quality competition among network transport providers. *Computational Management Science*, 11(4):475–502, 2014.

[13] A. Nagurney and D. Zhang. *Projected Dynamical Systems and Variational Inequalities with Applications*. Kluwer Academic Press, 1996.

[14] Anna Nagurney, Min Yu, and Jonas Floden. Supply chain network sustainability under competition and frequencies of activities from production to distribution. *Computational Management Science*, 10(4):397–422, 2013.

[15] W. O. Paradis and D. D. Perlmutter. Tracking function approach to practical stability and ultimate boundedness. *AIChE Journal*, 12(1):130–136, 1966.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] DavidH. Raab and EdwardH. Green. A cosine approximation to the normal distribution. *Psychometrika*, 26(4):447–450, 1961.

[18] Marco Sandri. Numerical calculation of Lyapunov exponents. *Mathematica Journal*, 6(3):78–84, 1996.

[19] Gesualdo Scutari, Daniel Palomar, Francisco Facchinei, and Jong-shi Pang. Convex optimization, game theory, and variational inequality theory. *IEEE Signal Processing Magazine*, 27(3):35–49, May 2010.

[20] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.

[21] Yue Wang, Alexandra Meliou, and Gerome Miklau. A consumer-centric market for database computation in the cloud. Technical report, University of Massachusetts, 2015.

[22] Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano. Determining lyapunov exponents from a time series. *Physica*, pages 285–317, 1985.