Economics Bulletin

Volume 40, Issue 2

A geometric programming approach to dynamic economic models

Inna Tsener
Universitat de les Illes Balears

Abstract

Geometric programming (GP) has several attractive features: it is tractable in large-scale problems, requires no initial guess or tuning of solver parameters, guarantees the convergence to a global optimum and can deal with kinks. In this note, I argue that GP is a potentially promising tool in economics. First, I show that a stylized finite-horizon growth model can be mapped into a GP format by using simple transformations. Second, I show that GP methods produce accurate and reliable solutions including the case of occasionally binding constraints which cannot be easily treated with conventional solvers. Examples of MATLAB codes are provided.

This article is based on a chapter of my PhD thesis written at the University of Alicante - I thank Lilia Maliar and Serguei Maliar for their guidance. I acknowledge the support of the University of Alicante and the University of the Balearic Islands and I thank Kenneth L. Judd and anonymous referees for many valuable comments.
Citation: Inna Tsener, (2020) "A geometric programming approach to dynamic economic models", *Economics Bulletin*, Volume 40, Issue 2, pages 1068-1074
Contact: Inna Tsener - inna.tsener@uib.es.
Submitted: January 18, 2019. **Published:** April 17, 2020.

1. Introduction

Dynamic models arise in every field of economics and there is a variety of numerical methods which can be used to find their optimal solutions; see Judd (1998), Rust (2008), Maliar and Maliar (2014) and Fernández-Villaverde et al. (2016) for reviews. In this note, I argue that the geometric programming approach is a potentially promising alternative for analyzing dynamic economic models. Geometric programming (GP) is a field of mathematical optimization which studies problems in which the objective function and constraints are formulated in terms of monomial and posynomial functions. The main motivation for using geometric programming in engineering is the efficiency and robustness with which geometric programs can be solved. Interior-point algorithms developed for geometric programming solve large problems very quickly, require no initial guess or tuning of solver parameters, guarantee the convergence to a global optimum and can deal with kinks and nondifferentiabilities; see Boyd et al. (2007).

To illustrate the geometric programming approach, I use a simple finite horizon deterministic neoclassical growth model with non-negative constraint on investment. In general, such a model does not have the exact GP representation, however, I show that it can still be well approximated by a geometric program. I assess the numerical properties of the solution delivered by geometric programming methods, and I find that they produce accurate and reliable solutions to models, including those with kinks and nondifferentiabilities which cannot be easily treated with conventional solvers.

My analysis focuses on a deterministic version of the model, however, it can be easily extended to stochastic problems by using a certainty equivalence approach, which essentially requires constructing multiple deterministic paths under different realizations of shocks. In particular, the proposed GP method can be used in the context of path solving methods such as those of Fair and Taylor (1983), Grüne et al. (2015) and Cai et al. (2017); including challenging nonstationary and unbalanced growth problems that cannot be studied with conventional solution methods; see Maliar et al. (2015) for discussion of such problems.

The remainder of this note is organized as follows. In Section 2, I describe a geometric program. In Section 3, I show how the neoclassical growth model can be reformulated as a geometric program. In Section 4, I provide numerical results and finally, in Section 5 I conclude.

2. Geometric Program

A geometric program is an optimization problem specified in terms of monomial and posynomial functions.

Definition 1 *(Monomial function) A function $f : \mathbb{R}^n_{++} \to \mathbb{R}_{++}$ is a monomial if it can be written as:*

$$f(x_1, \dots, x_n) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n},$$

where $c > 0$ and $a_i \in \mathbb{R}, i = 1, \dots, n$ are coefficients and exponents of the monomial, respectively.

Examples of monomial functions in economics are a Cobb-Douglas production function and a utility function, accordingly,

$$f(k,l) = k^{\alpha} l^{1-\alpha} \qquad \text{and} \qquad u(c,l) = \frac{(c^{\mu}(1-l)^{1-\mu})^{1-\eta}}{1-\eta},$$

where c, k and l, are consumption, capital and labor, respectively; $\alpha, \mu, \eta \in (0,1)$.

Definition 2 *(Posynomial function) A function $f : \mathbb{R}^n_{++} \to \mathbb{R}_{++}$ is called a posynomial if f is a finite sum of monomials on $\mathbb{R}^n_{++}$:*

$$f(x_1, \ldots, x_n) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}},$$

where $c_k > 0$ and $a_{ik} \in \mathbb{R}, i = 1, \ldots, n, k = 1, \ldots, K$ are coefficients and exponents of the posynomial, respectively.

Posynomials differ from polynomials as the exponents of a polynomial must be nonnegative integers while its coefficients and variables can take on any real number. In contrast, posynomial's exponents can take on any real number while its coefficients and variables must be positive.

Posynomial functions can often be found in economic models, for example,

$$c + i \qquad \text{and} \qquad f(k,l) + (1-\delta)k,$$

where i is investment and $\delta \in (0,1)$ is depreciation rate.

Definition 3 *(Geometric program) A geometric program is an optimization problem of the form:*

$$\min f_0(\mathbf{x}) \tag{1}$$

$$\begin{aligned} s.t.\ f_i(\mathbf{x}) &\leq 1, \quad i = 1, ..., m, & (2)\\ g_i(\mathbf{x}) &= 1, \quad i = 1, ..., p, & (3) \end{aligned}$$

where $\mathbf{x}$ is a vector of optimization variables, $f_i(\mathbf{x})$ are posynomial functions and $g_i(\mathbf{x})$ are monomials.

The optimization problem (1)–(3) is known in the literature as a geometric program in standard form. A distinctive feature of GP analysis is that the optimization variables in vector $\mathbf{x}$ are restricted to be positive by construction – this is a useful property for economic problems with kinks and inequality constraints. Many optimization problems can be either reduced to or approximated by the form (1)–(3); for examples in engineering and economics see Brightler and Philips (1976), Boyd et al. (2007) and Liu (2006).

Geometric programs are solved by interior point methods for convex optimization problems. Such methods does not require providing initial guess or tuning of solver parameters and can find a global optimum of a large GP problem quickly. Efficient primal-dual interior point methods for GPs are available in the MOSEK software package, the Python package GPkit, and the MATLAB toolboxes CVX and GGPLAB.

3. Formulating the neoclassical growth model as a GP

To illustrate the geometric programming approach, I consider a simple finite horizon deterministic neoclassical growth model with non-negative constraint on investment

$$\max_{\{k_{t+1},c_t\}_{t=0}^{T}} \sum_{t=0}^{T} \beta^t \frac{c_t^{1-\gamma}-1}{1-\gamma} \tag{4}$$

$$\text{s.t. } c_t + k_{t+1} \leq (1-\delta)\, k_t + f\,(z_t, k_t)\,, \tag{5}$$

$$(1-\delta)\, k_t \leq k_{t+1}, \tag{6}$$

where $c_t > 0$ and $k_t > 0$ denote consumption and capital, respectively; z_t is the productivity level and a representative agent has perfect foresight about its future evolution; $f\,(z_t, k_t) = z_t k_t^{\alpha}$ is the Cobb-Douglas production function with $\alpha \in (0,1)$; $\beta \in (0,1)$ is the discount factor; $\delta \in [0,1]$ is the depreciation rate; initial and terminal conditions for capital are given, k_0 and k_{T+1}, respectively. I solve for paths of capital and consumption that maximize the objective function (4) and satisfy the budget constraint (5) and the non-negative constraint on investment (6).[1]

In general, problem (4)–(6) is not a GP in standard form: first, a representative agent maximizes its utility while the geometric program in definition (3) requires minimization; second, the utility function is generally not a posynomial; and finally, the left and right hand sides of (5) are posynomials and their ratio is not a posynomial.

Nonetheless, problem (4)–(6) can be approximated by a geometric program using simple transformations and term rearrangements. As an example, consider a version of the model with the coefficient of risk aversion greater than one. First, when $\gamma > 1$, the objective function in (4) is a sum of monomials of which the coefficients are negative because $1-\gamma < 0$. I reformulate the problem of a representative agent to minimize $\sum_{t=0}^{T} \beta^t c_t^{1-\gamma}(\gamma-1)^{-1}$ and the resulting objective function becomes a posynomial[2]. Second, to construct an approximation of the budget constraint (5) I use a condensation method. The basic underlying principle of condensation is to construct a monomial approximation to a posynomial function by using a set of auxiliary weights. The right hand side of the resource constraint (5) is a posynomial and in each time period it can be approximated by the monomial function

$$M(k_t) \doteq \left(\frac{(1-\delta)k_t}{\omega_{1,t}}\right)^{\omega_{1,t}} \left(\frac{z_t k_t^{\alpha}}{\omega_{2,t}}\right)^{\omega_{2,t}},$$

where

$$\omega_{1,t} = \frac{(1-\delta)k_t}{z_t k_t^{\alpha} + (1-\delta)k_t} \quad \text{and} \quad \omega_{2,t} = \frac{z_t k_t^{\alpha}}{z_t k_t^{\alpha} + (1-\delta)k_t} \tag{7}$$

are positive weights, $\sum_{i=1}^{2} \omega_{i,t} = 1, t = 0, ..., T$. This allows me to rewrite the resource constraint in a way suitable for geometric programming

$$c_t M(k_t)^{-1} + k_{t+1} M(k_t)^{-1} \leq 1. \tag{8}$$

[1] The turnpike literature shows that a solution to a finite horizon economy approximates well the solution to an infinite horizon economy in the first periods as time horizon increases, see McKenzie (1976). Hence, the results of the discussion presented in this note also apply to the infinite horizon economies.

[2] Note that a constant term $\sum_{t=0}^{T} \beta^t/(1-\gamma)$ is dropped since it does not affect the optimal solution.

The non-negative constraint on investment can be trivially reformulated in terms of monomial function

$$(1-\delta)k_t k_{t+1}^{-1} \leq 1. \tag{9}$$

Using the simple transformations described above we obtain a geometric program

$$\min_{\{k_{t+1},c_t\}_{t=0}^T} \sum_{t=0}^{T} \beta^t \frac{c_t^{1-\gamma}}{\gamma-1} \tag{10}$$

$$\text{s.t. } c_t M(k_t)^{-1} + k_{t+1} M(k_t)^{-1} \leq 1, \tag{11}$$

$$(1-\delta)k_t k_{t+1}^{-1} \leq 1. \tag{12}$$

The procedure which finds an accurate approximated solution to the original model consists of three steps: Step 1, guess a sequence of weights $\{\omega_{1,t}, \omega_{2,t}\}_{t=0}^T$; Step 2, solve the optimization problem formed by (10)–(12); Step 3, using the solution obtained on Step 2 and equations in (7) recompute the weights and return to Step 1. The iteration stops when the difference between the weights from two successive iterations is small. By iterating on weights $\{\omega_{1,t}, \omega_{2,t}\}_{t=0}^T$ one can solve the condensed problem (10)–(12) and obtain a solution which is in the feasible set of solutions to the original problem. Rosenberg (1979) shows that the condensation method converges from any arbitrary initial guess to a solution under mild conditions.

4. Numerical Analysis

In this section, I evaluate the performance of the interior-point methods developed for geometric programming in the context of the finite horizon growth model.

4.1. Methodology

I parametrize model (4)–(6) with $\gamma = \{0.1, 1, 10\}$, $\alpha = 0.36$, $\beta = 0.99$, $\delta = 0.025$, $T = \{50, 100, 150\}$. The productivity level follows the AR(1) process with $\rho = 0.95$ and $\sigma = \{0.02, 0.2\}$. To solve GPs I use a MATLAB based toolbox GGPLAB for specifying and solving geometric programs. I compare the solution computed by geometric programming techniques to the *benchmark* solutions obtained by conventional nonlinear solution methods. Specifically, I use Matlab's function "fmincon" to construct benchmark solution. The numerical results are not sensitive to a specific choice of the solver. I use MATLAB R2014b software on a MacBook Pro laptop with Intel Core i7 (2,9GHz) and 8 GB RAM. Examples of MATLAB codes are provided on https://sites.google.com/site/innatsener/codes.

4.2. Results

In the left and right panels of Figure 1, I plot the solution for investment produced by solving the neoclassical growth model with slack and occasionally binding non-negative constraint on investment, respectively. For the model with occasionally binding investment constraint, I construct only the GP solution but not the benchmark solution as the standard numerical solvers are not designed to handle problems with kinks.

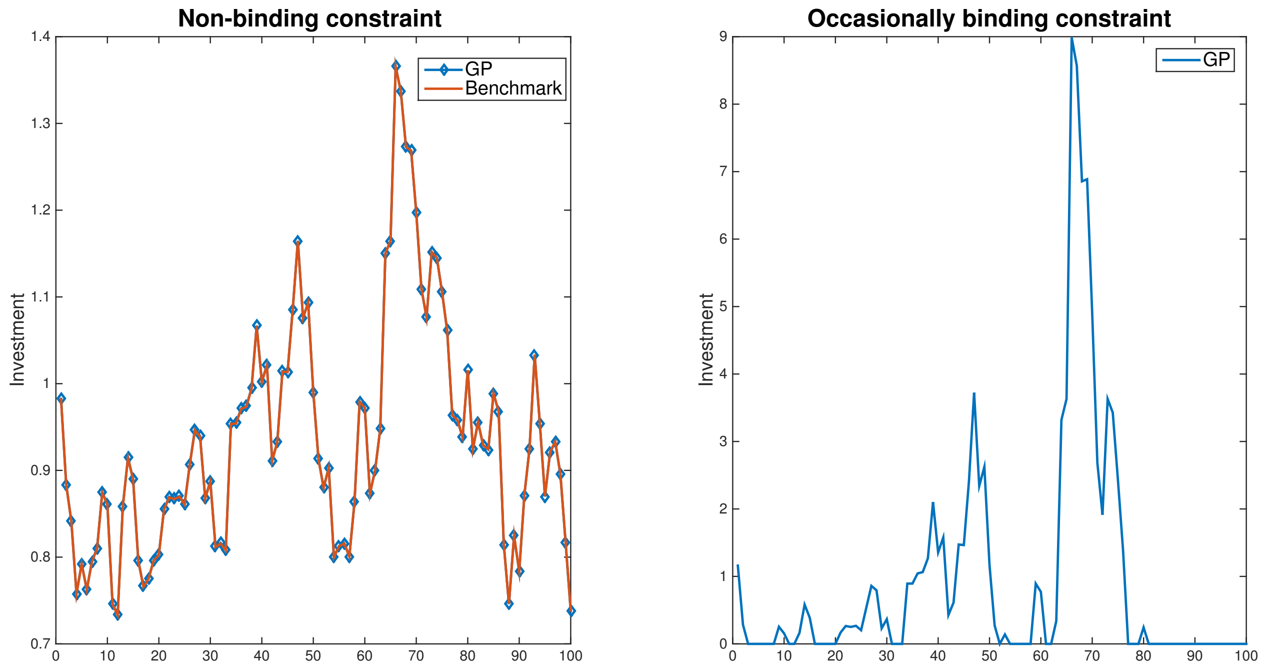

Figure 1: Investment path under slack and binding constraint

My main result in the left panel is that the differences between the solution of the condensed geometric program and the benchmark solution are negligible. Here, investment series take on positive values in all periods which implies that the investment constraint is always slack. This particular solution was obtained by setting $\gamma = 1$ and $\sigma = 0.02$. Table 1 reports mean and maximum (unit-free) absolute percentage differences between the two solutions for different values of risk-aversion and different time horizons. As we can see, the average percentage differences between two solutions for capital stay within a range of 0.0009% to 0.05%, while the maximum differences reach 0.2%. The average and maximum percentage differences between the two solutions for investment are no higher than 0.6% and 3%, respectively, which is about the accuracy level of various solvers.

For the model with slack investment constraint, the running time for constructing both the GP and benchmark solutions is about 15 seconds for $T = 100$. Thus, the GP method can find a global optimum of dynamic economic models at a relatively modest cost. In general, constructing a longer path requires a higher running time, however, for lower values of the risk aversion parameter the running time is under a minute in all my experiments. Generally, I find that solving a model with an occasionally binding constraint is slightly more expensive than solving one without it.

In the right panel of Figure 1, I consider $\gamma = 1$ and high volatility of $\sigma = 0.2$. As a result of this change the investment constraint binds occasionally because when a large negative shock hits the economy, the representative agent consumes more of her capital stock in order to smooth her consumption over time. To be specific, the investment hits the lower bound in roughly fifty percent of times. Hence, this result suggests that geometric programming is

Table 1: Accuracy of solution obtained by geometric programming

γ	T	Capital e_{mean}	Capital e_{max}	Investment e_{mean}	Investment e_{max}	CPU
0.1	50	4.81(−2)	1.02(−1)	0.63	3.23	19
	100	5.03(−2)	1.42(−1)	0.64	2.57	40
	150	4.24(−2)	1.46(−1)	0.58	1.87	60
1	50	1.18(−2)	1.91(−2)	3.30(−2)	5.60(−2)	5
	100	3.14(−2)	5.04(−2)	5.31(−2)	9.29(−2)	10
	150	1.51(−2)	3.35(−2)	2.57(−2)	5.92(−2)	18
10	50	0.09(−2)	0.15(−2)	0.27(−2)	0.65(−2)	11
	100	0.25(−2)	0.42(−2)	0.54(−2)	1.49(−2)	60
	150	1.21(−2)	1.96(−2)	1.73(−2)	4.13(−2)	126

Notes: e_{mean} and e_{max} are, respectively, mean and maximum absolute percentage differences between the solution to the condensed geometric problem and the benchmark solution; γ is the risk aversion parameter; T is the time horizon; CPU is the time necessary to compute the solution (in seconds); $\zeta(-j)$ represents $\zeta \times 10^{-j}$.

capable of approximating a solution to highly nonlinear finite horizon economic models such as this one.

5. Conclusion

In this note, I show how a simple finite horizon neoclassical growth model with non-negative constraint on investment can be mapped into a geometric program and I evaluate numerical properties of the solution obtained by geometric programming methods. I find that in the absence of borrowing constraints, GP methods deliver essentially the same solution as standard conventional solvers. In addition however, they are computationally stable when solving the problems with kinks and nondifferentiabilitites. The proposed GP method can be used in the context of Fair and Taylor (1983) and other similar path-solving methods.

The results shown in this note are preliminary. Extending the geometric programming approach to more sophisticated and high-dimensional economic problems and evaluating how numerical costs of GP methods change with the dimensionality of the problem is an interesting direction for future research.

References

Beightler, C.S. and Philips, D.T. (1976) “Applied geometric programming”, John Wiley and Sons: New York.

Boyd, S., Kim, S.J., Vandenberghe, L. and Hassibi, A. (2007) “A tutorial on geometric programming” *Optimization and Engineering* **8**, 67–127.

Cai, Y., Judd, K. and Steinbuks, J. (2017) "A nonlinear certainty equivalent approximation method for dynamic stochastic problems" *Quantitative Economics* **8(1)**, 117–147.

Fair, R. and Taylor, J.B. (1983) "Solution and Maximum Likelihood Estimation of Dynamic Nonlinear Rational Expectations Models" *Econometrica* **51(4)**, 1169–1185.

Fernández-Villaverde, J., Rubio-Ramírez, J.F. and Schorfheide, F. (2016) "Solution and Estimation Methods for DSGE Models" in *Handbook of Macroeconomics* by J.B. Taylor and H. Uhlig, Eds., Elsevier, pp. 527–724.

Grüne, L., Semmler, W. and Stieler, M. (2015) "Using nonlinear model predictive control for dynamic decision problems in economics" *Journal of Economic Dynamics and Control* **60**, 112–133.

Judd, K. (1998) "Numerical methods in economics", MIT Press:Cambridge, MA.

Liu, S.T. (2006) "A geometric programming approach to profit maximization" *Applied Mathematics and Computation*, **182**, 1093–1097.

Maliar, L., Maliar, S., Taylor, J. and Tsener, I. (2015) "A tractable framework for analyzing a class of nonstationary markov models" NBER Working Paper 21155.

Maliar, L. and Maliar, S. (2014) "Numerical Methods for Large-Scale Dynamic Economic Models" in *Handbook of Computational Economics* by K. Schmedders and L. Judd, Eds., Elsevier Science: Amsterdam, pp. 325–477.

McKenzie, L. (1976) "Turnpike theory" *Econometrica* **44**, 841–865.

Rosenberg, E. (1979) "A Globally Convergent Condensation Method for Geometric Programming" Technical report Sol-79-11

Rust, J. (2008)"Dynamic Programming" in *The New Palgrave Dictionary of Economics* by S. Durlauf and L. Blume, Eds., Palgrave Macmillian.