# A Stochastic Simulation based Algorithm for Solving Dynamic Economic Models*

Srinivasan Murali †

June 8, 2015

**Abstract**

One of the most important bottlenecks in writing elaborate dynamic economic models is the curse of dimensionality. It has been shown that stochastic simulation framework helps in relaxing this curse. I develop an algorithm using this framework and demonstrate its scalability and robustness by solving a multi-country business cycle model with 12 continuous state variables. I also extend this algorithm for solving discrete choice dynamic programming problems by attempting to solve Khan and Thomas (2003) lumpy investment model and Arellano (2008) sovereign default model.

JEL Classification: C63,C68.

## 1 Introduction

We assume economic agents have expectations on their future and model their decision making as a dynamic optimization problem. Even though dynamic economic models are the mainstay of macroeconomic research, it is a rarity that these dynamic programming problems have analytical solutions. Hence, we have to resort to algorithms to solve these models numerically. One of the major bottlenecks faced by researchers in numerically solving these problems is the curse of dimensionality - computational time or memory required to solve a problem increases exponentially in the number of dimensions of the state space.

*I am grateful to Aubhik Khan, Julia Thomas and Jaromir Nosal for helpful comments and advice. I also thank the participants at Khan-Thomas Workshop and macro lunch for their suggestions.

†The Ohio State University; `murali.20@osu.edu`

Approximate Dynamic Programming(ADP) is a framework developed primarily in the fields of engineering and operations research to tackle the problem of curse of dimensionality. This literature attempts to find approximate solutions to high dimensional problems by employing stochastic simulation and learning algorithms. Bertsekas and Tsitsiklis (1996) gave the first theoretical foundations for this approach, while Van Roy et al. (1997) was the first application of an ADP algorithm to a real world problem. This paper attempted to solve a retail optimization problem with 33 state variables. Powell (2011) and Buşoniu et al. (2010) collate different applications and practical issues in applying these algorithms in the fields of operations research and reinforcement learning respectively.

Stochastic simulation based methods are not new to economics literature. Parameterized expectations approach introduced in Marcet (1991) and Den Haan and Marcet (1990) solves for the equilibrium quantities by approximating the conditional expectation of euler equations using exponentiated polynomials and using simulations and nonlinear regression to update the approximation[1]. Judd et al. (2011) also uses euler equation as the basis for simulation, but parameterizes policy function instead of conditional expectations. This paper also documents various regularization techniques that can be employed to stabilize the stochastic simulation framework. Arellano et al. (2014) uses envelope condition as the basis for their solution algorithm instead of euler equations.

The solution method in this paper uses the dynamic programming approach by parameterizing the continuation value of the functional equation and uses stochastic simulation methods to update the approximation[2]. I apply this method to solve the lumpy investment model of Khan and Thomas (2003) where firms face a nontrivial discrete choice problem. Even though the value function of this model is not smooth, my method can still be used because the continuation value is a smooth function. I also use this algorithm to solve a multi-country business cycle model with 12 continuous states. The results show that, the algorithm performs very well in higher dimensions both in terms of computational time and accuracy. Finally, I apply this algorithm to a sovereign default model of Arellano (2008) whose value function and continuation value are not smooth.

[1] Heterogeneous agent models like Krusell et al. (1998) and Khan and Thomas (2003) uses stochastic simulation to obtain the aggregate law of motion consistent with the already solved value function.

[2] Hull (2015) also uses stochastic simulations in a dynamic programming setting, but restricts the states and the chosen controls to a pre-specified grid, which is not the case in my algorithm.

The rest of the paper is organized as follows: In section 2, I introduce a generic dynamic optimization problem. Section 3 states the baseline algorithm used in this paper. Section 4 applies the algorithm to solve a neoclassical growth model, multi-country business cycle model, lumpy investment model and the sovereign default model. In section 5, I conclude.

## 2 A Dynamic Optimization Problem

Let $x_t$ denote the vector of endogenous state variables while $z_t$ be the vector of exogenous state variables at period $t$. The economic agent at any period $t$, after observing $\{x_t, z_t\}$ maximizes his/her instantaneous utility $F(x_t, x_{t+1}, z_t)$ and discounted future lifetime utility by choosing the next period's endogenous state vector $x_{t+1}$ subject to the constraint that $x_{t+1}$ lies in the set $\Gamma(x_t, z_t)$.

$$v(x_t, z_t) = \max_{x_{t+1} \in \Gamma(x_t, z_t)} \left( F(x_t, x_{t+1}, z_t) + \beta \int v(x_{t+1}, z_{t+1}) Q(z_t, dz_{t+1}) \right) \tag{1}$$

where $v$ represents the optimal value of the lifetime utility, $\beta$ the discount factor used by the agent to discount the future utility and $Q$ is the probability distribution of tomorrow's exogenous state $z_{t+1}$ conditional on today's state $z_t$.

The policy function $g(x_t, z_t)$ gives the optimal choice of $x_{t+1}$ given the current period's state variables $\{x_t, z_t\}$.

$$g(x_t, z_t) = \underset{x_{t+1} \in \Gamma(x_t, z_t)}{\arg\max} \left( F(x_t, x_{t+1}, z_t) + \beta \int v(x_{t+1}, z_{t+1}) Q(z_t, dz_{t+1}) \right) \tag{2}$$

Before proceeding further, let us define *continuation value* of the above problem. The continuation value at the end of period $t-1$ (represented as $v^u(x_t, z_{t-1})$), is defined as the expected value of being in the state $(x_t, z_t)$ where the expectation is taken over the unobserved $z_t$ conditional on the current exogenous state $z_{t-1}$.

$$v^u(x_t, z_{t-1}) = \int v(x_t, z_t) Q(z_{t-1}, dz_t) \tag{3}$$

We can reformulate the original optimization problem (1) in terms of the continuation value as follows.

$$
\begin{aligned}
v^u(x_t, z_{t-1}) &= \int v(x_t, z_t) Q(z_{t-1}, dz_t) \\
&= \int \left[ \max_{x_{t+1} \in \Gamma(x_t, z_t)} \left( F(x_t, x_{t+1}, z_t) + \beta v^u(x_{t+1}, z_t) \right) \right] Q(z_{t-1}, dz_t)
\end{aligned}
\tag{4}
$$

The corresponding policy function is given by

$$
g(x_t, z_t) = \underset{x_{t+1} \in \Gamma(x_t, z_t)}{\arg\max} \left( F(x_t, x_{t+1}, z_t) + \beta v^u(x_{t+1}, z_t) \right) \tag{5}
$$

For the baseline algorithm, I use the above formulation of the problem in terms of the continuation value. The advantage of this formulation is, the policy function no longer has any expectation operator and hence analytical methods can be used to solve for the policy once we assume a suitable functional form for the continuation value.

## 3 The Algorithm

This algorithm combines the traditional value function iteration and simulation methods to solve the problem defined in the previous section. We start with an initial functional approximation for the continuation value and initial values for the exogenous and endogenous states. Simulate the exogenous state for a finite number of periods. We then simulate the model using the policy function (equation (5)) to obtain the next period's endogenous state given the current period's state and the continuation value. Continue this simulation till we obtain sufficient data points to update the estimate of the implied continuation value. The procedure continues until there is no change in the functional approximation across iterations. Formally,

- Assume a functional form for continuation value($v^u$) and the corresponding coefficients($\theta_v$).
- Given the initial exogenous state($z_0$), simulate the realizations $\{z_t\}_{t=1}^T$ for $T$ periods.
- While ($\theta_v$ doesn't converge)
  - for $t = 1$ to T
    - Obtain $x_t = g(x_{t-1}, z_{t-1})$ using (5).
    - Calculate the continuation value ($v_t^u$) at the state $(x_t, z_{t-1})$ from (4).

- End for
- Regress $\{v^u\}_{t=1}^{T}$ on $\{x_t, z_{t-1}\}_{t=1}^{T}$ to obtain the updated coefficients $\hat{\theta_v}$.
- New coefficient $(\theta_v) = (1-\eta)\theta_v + \eta\hat{\theta_v}$.

- End while

The algorithm dynamically determines the region of state space in which the solution lies and solves for the continuation value and the policy function in that region. This obviates the need to define a region of approximation prior to the start of the algorithm which is common in other classes of solution methods. This algorithm employs regression techniques to estimate the continuation value compared to the more common interpolation techniques used by the grid based solution methods.

In the next section, we will show how this algorithm can be used to solve a wide variety of dynamic economic models along with the performance of the algorithm in different applications. All models are solved using MATLAB R2014a in my laptop with Intel(R) Core(TM) i5-4200 CPU (1.6 GHz) and 8GB RAM.

# 4 Applications

## 4.1 Neoclassical Growth Model

The first application is the workhorse model of macroeconomic research. This model has a representative agent whose objective is to maximize his expected discounted utility over an infinite lifetime. At the start of each period, the agent observes the realization of productivity($z_t$) and capital($k_t$) and decides how much to consume and save in order to achieve his goal. The production next period is determined by his savings this period which in turn will affect his consumption and savings next period.

$$
\begin{aligned}
v(z_t,k_t) &= \max_{c_t,k_{t+1}} \left( \frac{c_t^{1-\sigma}-1}{1-\sigma} + \beta \mathrm{E}[v(z_{t+1},k_{t+1})|z_t] \right) \\
\text{subject to} & \\
c_t + k_{t+1} &\leq z_t k_t^{\alpha} + (1-\delta)k_t \\
c_t &\geq 0, k_{t+1} \geq 0 \\
z_{t+1} &= z_t^{\rho} e^{\epsilon_{t+1}},\ \epsilon_{t+1} \sim N(0,\sigma_\epsilon^2) \\
z_0,k_0 \text{ given} & \qquad (6)
\end{aligned}
$$

We can rewrite the model in terms of continuation value($v^u$) as follows:

$$
\begin{aligned}
v^u(z_{t-1},k_t) &= \mathrm{E}\left[ \max_{c_t,k_{t+1}} \left( \frac{c_t^{1-\sigma}-1}{1-\sigma} + \beta v^u(z_t,k_{t+1}) \right) \middle| z_{t-1} \right] \\
\text{subject to} & \\
c_t + k_{t+1} &\leq z_t k_t^{\alpha} + (1-\delta)k_t \\
c_t &\geq 0, k_{t+1} \geq 0 \\
z_t &= z_{t-1}^{\rho} e^{\epsilon_t},\ \epsilon_t \sim N(0,\sigma_\epsilon^2) \\
z_0,k_0 \text{ given} & \qquad (7)
\end{aligned}
$$

The policy function for the model can be given by:

$$
g(z_t,k_t) = \underset{k_{t+1}\in[0,z_t k_t^{\alpha}+(1-\delta)k_t]}{\arg\max} \left( \frac{(z_t k_t^{\alpha} + (1-\delta)k_t - k_{t+1})^{1-\sigma}-1}{1-\sigma} + \beta v^u(z_t,k_{t+1}) \right) \qquad (8)
$$

For this exercise, I approximate the continuation value with a complete cubic polynomial of the state variables $z$ and $k$.

$$
v^u(z_{t-1},k_t) = \theta_1+\theta_2 z_{t-1}+\theta_3 z_{t-1}^2+\theta_4 z_{t-1}^3+\theta_5 k_t+\theta_6 k_t^2+\theta_7 k_t^3+\theta_8 z_{t-1}k_t+\theta_9 z_{t-1}^2 k_t+\theta_{10} z_{t-1}k_t^2 \qquad (9)
$$

By substituting (9) into (8), and under the assumption of $\sigma$ being 2, we can characterize the policy function with the following equation:

$$
ak_{t+1}^4 + (b-2ad)k_{t+1}^3 + (ad^2-2bd+c)k_{t+1}^2 + (bd^2-2cd)k_{t+1} + cd^2 - \frac{1}{\beta} = 0 \qquad (10)
$$

where $a, b, c$ and $d$ are functions of current states and approximation coefficients. In this case the policy function is given by a fourth order polynomial in $k_{t+1}$ and I calculate the eigenvalues of

| Parameters | Values | Meaning |
|---|---|---|
| $\alpha$ | 0.36 | Curvature of the production function |
| $\delta$ | 0.069 | Depreciation rate of capital |
| $\sigma$ | 2 | Coefficient of relative risk aversion |
| $\beta$ | 0.96 | Discount rate |
| $\rho$ | 0.859 | Persistence of productivity process |
| $\sigma_\epsilon$ | 0.014 | Standard deviation of shocks to productivity |
| $k_{ss}$ | 6.3161 | Steady-state capital |
| $T$ | 1000 | Length of each simulation |
| $\eta$ | 0.3 | Learning rate |

Table 1: Neoclassical Growth Model - Parameters

| Algorithm | Run Time(secs) | Max. Error | Avg. Error |
|---|---|---|---|
| Grid | 20 | 8.2998e-05 | 3.8346e-06 |
| Simulation | 225 | 2.8890e-04 | 3.6176e-05 |

Table 2: Neoclassical Growth Model - Performance

the companion matrix to solve for $k_{t+1}$. Armed with this policy function, we use the algorithm from the previous section to solve the growth model.

Table 1 lists the parameter values that have been used to solve this model. All the parameter values are pretty standard except for the algorithm specific parameters. $T$ is the total number of periods the model is simulated in each iteration and is set to 1000. I have tried multiple values while this number gives higher stability and accuracy. The other parameter $\eta$ is the rate at which the algorithm learns about the new coefficient. Higher the learning rate, the faster is the rate of convergence of the algorithm but lesser is its stability. I find 0.3 to be the highest value of $\eta$ that guarantees convergence of the algorithm for this model.

The performance of this algorithm is compared to a value function iteration algorithm where the value function is represented on a grid of $z$ and $k$. The space of productivity realization is discretized into $N_z$ (here, 5) states by employing Tauchen (1986) while the space of capital $k$ is discretized linearly into $N_k$ (here, 11) states. The value function is represented on a grid of $N_z \times N_k$ while the continuation value is interpolated using piecewise polynomial cubic splines.

Table 2 gives the runtime and the euler equation error statistics of the simulation based

| Algorithm | Productivity | Capital | Consumption | Output | Investment |
|---|---|---|---|---|---|
| *Mean* | | | | | |
| Grid | 0.9995 | 6.3133 | 1.5046 | 1.9403 | 0.4357 |
| Simulation | 0.9996 | 6.3154 | 1.5050 | 1.9407 | 0.4357 |
| *Standard Deviation* | | | | | |
| Grid | 0.0274 | 0.0052 | 0.0057 | 0.0136 | 0.0421 |
| Simulation | 0.0275 | 0.0051 | 0.0056 | 0.0133 | 0.0409 |

*standard deviation of log data HP filtered with parameter value of 100.*

Table 3: Neoclassical Growth Model - Business Cycle Moments

algorithm and the benchmark algorithm. The grid based algorithm does a much better job both in terms of accuracy and runtime. It is understandable because simulation based algorithms are inherently costly as we need to simulate a lot of datapoints to achieve accuracy and stability. So we need a bigger model to completely leverage the true potential of this algorithm, which is where we turn to next. Before we turn to the next application, table 3 compares the business cycle moments implied by both the algorithms. As seen from the table, both the algorithms generate statistics that are very close to each other.

### 4.2 Multi-Country Real Business Cycle Model

This is one of the models studied in the February 2011's Journal of Economic Dynamics and Control(JEDC) computational suite project comparing different solution methods. Haan et al. (2011) provides more details on this model. It is a multi-country extension of the model solved in the previous section. In this model, there are $N$ countries each facing both idiosyncratic shocks (shocks that are specific to a particular country) and an aggregate shock (shock that is common to all the countries). There is a global planner whose objective is to maximize the weighted average of utility of $N$ countries. The planner observes the productivity realization and capital level of each country at the start of a period and decides the consumption and savings of each country to achieve her objective.

$$v(z_t, k_t) = \max_{\{c_{j,t}, k_{j,t+1}\}_{j=1}^{N}} \left( \sum_{j=1}^{N} \tau_j log(c_{j,t}) + \beta \mathrm{E}[v(z_{t+1}, k_{t+1})|z_t] \right)$$

subject to

$$\sum_{j=1}^{N}\left(c_{j,t}+k_{j,t+1}+\frac{\phi}{2}\frac{(k_{j,t+1}-k_{j,t})^2}{k_{j,t}}\right)=\sum_{j=1}^{N}(Az_{j,t}(k_{j,t})^{\alpha}+k_{j,t})$$

$$logz_{j,t}=\rho logz_{j,t-1}+\sigma(e_t+e_{j,t});\ \ e_t,e_{j,t}\sim N(0,1)$$

$$z_t=\{z_{j,t}\}_{j=1}^{N};k_t=\{k_{j,t}\}_{j=1}^{N}$$

The state vector of the model consists of productivity realizations and capital stock of $N$ countries. The productivities of the countries are correlated with each other and each country is subjected to a quadratic capital adjustment cost. This cost is important for our purpose because the adjustment cost forces the planner's decision to be dependent on the entire distribution of capital and productivity realizations across the world, instead of just the aggregate levels of capital and productivity.

As before, let us reformulate the above problem in terms of the continuation value. Let $v^u(z_{t-1},k_t)$ represent the continuation value of the planner's problem at period $t-1$.

$$v^u(z_{t-1},k_t)=\mathrm{E}\left[\max_{\{c_{j,t},k_{j,t+1}\}_{j=1}^{N}}\left(\sum_{j=1}^{N}\tau_j log(c_{j,t})+\beta v^u(z_t,k_{t+1})\right)\middle| z_{t-1}\right]$$

subject to

$$\sum_{j=1}^{N}\left(c_{j,t}+k_{j,t+1}+\frac{\phi}{2}\frac{(k_{j,t+1}-k_{j,t})^2}{k_{j,t}}\right)=\sum_{j=1}^{N}(Az_{j,t}(k_{j,t})^{\alpha}+k_{j,t})$$

$$logz_{j,t}=\rho logz_{j,t-1}+\sigma(e_t+e_{j,t});\ \ e_t,e_{j,t}\sim N(0,1)$$

$$z_t=\{z_{j,t}\}_{j=1}^{N};k_t=\{k_{j,t}\}_{j=1}^{N}$$

Let $g$ represent the policy correspondence associated with this model. It gives the vector of next period's capital stock $\{k_{j,t+1}\}_{j=1}^{N}$ given the current state $\{z_t,k_t\}$.

$$g(z_t,k_t)=\underset{\{c_{j,t},k_{j,t+1}\}_{j=1}^{N}}{\arg\max}\left(\sum_{j=1}^{N}\tau_j log(c_{j,t})+\beta v^u(z_t,k_{t+1})\right) \tag{11}$$

With a suitable parametric approximation of the continuation value, we can solve the model using the algorithm introduced before. Before we go into the exact details of solving the model, let us discuss how the policy correspondence looks when $N$ is 2. This restriction makes the discussion easier and can be easily extended to $N$ more than 2.

The continuation value is approximated using a complete quadratic polynomial of the state vector.

$$\begin{aligned}
v^u(z_{t-1}, k_t) = \theta_1 + \theta_2 z_{1,t-1} + \theta_3 z_{2,t-1} + \theta_4 z_{1,t-1}^2 + \theta_5 z_{2,t-1}^2 + \\
\theta_6 k_{1,t} + \theta_7 k_{2,t} + \theta_8 k_{1,t}^2 + \theta_9 k_{2,t}^2 + \theta_{10} z_{1,t-1} z_{2,t-1} \\
+\theta_{11} k_{1,t} k_{2,t} + \theta_{12} z_{1,t-1} k_{1,t} + \theta_{13} z_{1,t-1} k_{2,t} + \theta_{14} z_{2,t-1} k_{1,t} + \\
\theta_{15} z_{2,t-1} k_{2,t}
\end{aligned} \tag{12}$$

Under this parameterization,the policy correspondence of a 2-country model can be given as follows.

$$\lambda_t c_{1,t} = \tau_1 \tag{13}$$

$$\lambda_t c_{2,t} = \tau_2 \tag{14}$$

$$\beta(\theta_6 + 2\theta_8 k_{1,t+1} + \theta_{11} k_{2,t+1} + \theta_{12} z_{1,t} + \theta_{14} z_{2,t}) = \lambda_t \left[1 + \phi \frac{k_{1,t+1} - k_{1,t}}{k_{1,t}}\right] \tag{15}$$

$$\beta(\theta_7 + 2\theta_9 k_{2,t+1} + \theta_{11} k_{1,t+1} + \theta_{13} z_{1,t} + \theta_{15} z_{2,t}) = \lambda_t \left[1 + \phi \frac{k_{2,t+1} - k_{2,t}}{k_{2,t}}\right] \tag{16}$$

$$\sum_{j=1}^{2} \left( c_{j,t} + k_{j,t+1} + \frac{\phi}{2} \frac{(k_{j,t+1} - k_{j,t})^2}{k_{j,t}} \right) = \sum_{j=1}^{2} (A z_{j,t} (k_{j,t})^\alpha + k_{j,t}) \tag{17}$$

Thus the first order conditions of a 2-country model consists of 5 equations to solve for 5 unknowns $\{c_{j,t}\}_{j=1}^2$,$\{k_{j,t+1}\}_{j=1}^2$ and $\lambda_t$ which is the Lagrange multiplier of the budget constraint. In general, a $N$-country model will have $2N + 1$ equations to solve for $2N + 1$ unknowns. So, even though it is possible to implement a value function iteration algorithm similar to the one used to solve the neoclassical growth model, solving this system of nonlinear equations every time to simulate the model can be very expensive. The cost increases rapidly as we increase the size of the model by adding more countries.

So, a policy iteration version of the original algorithm is used to solve this model. In this version, we explicitly parameterize the policy correspondence $g(z_t, k_t)$. The parametric approximation of the policy enables us to simulate the model at a very low cost. We update this policy approximation over the iterations and the algorithm continues till the policy parameters converge. This implementation of policy iteration algorithm follows from Buşoniu et al. (2010).

We start with an initial functional form for the policy and an initial value for the state vector. Simulate the exogenous state vector for a finite period using the given law of motion. With the assumed approximation for the policy, we can simulate the model forward to obtain the next period's endogenous state. Once we simulate enough data points, we find the continuation value ($v^u$) implied by the current policy approximation. This step is called *policy evaluation*. Once we obtain the implied continuation value, we can use this information to update the policy approximation. This step is called *policy improvement*. We iterate on this procedure with the updated policy until the policy approximation or the implied continuation value stops changing.

- Assume a functional form for the policy $g$ and continuation value $v^u$. Let the coefficient vector be $\theta_g$ and $\theta_v$ respectively.
- Given the initial exogenous state $z_0$, simulate the exogenous state realizations $\{z_t\}_{t=1}^T$ for T periods.
- while ($\theta_v$ doesn't converge)
  - for $t = 0$ to $T$
    - Obtain $k_{t+1} = g(k_t, z_t)$
  - End for
  - *Policy Evaluation:* Obtain the continuation value parameters($\theta_v$) implied by the policy $g$ using $\{z_t\}_{t=1}^T$ and $\{k_t\}_{t=1}^T$.
  - *Policy Improvement:* Update the coefficients of the policy($\theta_g$) using the continuation value approximation($\theta_v$) to obtain the updated policy $g$.
- End while

*Policy Evaluation*[3]

- Start with an initial guess for $\theta_v$.
- while ($\theta_v$ doesn't converge)
  - Calculate $\{v^u(z_t, k_{t+1})\}_{t=1}^T$ using $\{z_t\}_{t=1}^T$, $\{k_t\}_{t=1}^T$ and current approximation($\theta_v$) of the continuation value.

[3]One can also use finite horizon approximation to find the continuation value. We find this way to be unstable in this exercise. The choice of learning rate might help improve the stability.

– Obtain the updated continuation value($\hat{v_t^u}$) from the definition:

$$\hat{v_t^u} = \mathrm{E}\left[\left(\sum_{j=1}^{N} \tau_j log(c_{j,t}) + \beta v^u(z_t, k_{t+1})\right)\middle| z_{t-1}\right]$$

– Regress $\{\hat{v_t^u}\}_{t=1}^T$ on $\{z_t, k_{t+1}\}_{t=1}^T$ to obtain the updated coefficients $\hat{\theta_v}$

– $\theta_v = (1 - \eta_v)\theta_v + \eta_v \hat{\theta_v}$

- End while

This procedure is similar to the value function iteration but for a given policy $g$. So, no maximization is involved in calculating the continuation value in this step.

*Policy Improvement*

This step of the policy iteration algorithm uses the current approximation of the continuation value to update the policy approximation. In the context of 2-country model, the policy correspondence $g(z_t, k_t)$ gives the next period's capital for both the countries. So, $g(z_t, k_t) \equiv [g_1\ g_2](z_t, k_t)$ where $g_1(z_t, k_t) = k_{1,t+1}$ and $g_2(z_t, k_t) = k_{2,t+1}$. Equations (13) - (17) give the first order conditions for a 2-country model. The first order conditions will hold with equality only under the optimal policy. So, equations (15) and (16) can be used to update the arbitrary policy $g$ to obtain the new policy $\hat{g}$ as follows[4].

$$\hat{g_1}(z_t, k_t) = \left[\frac{\beta(\theta_6 + 2\theta_8 g_1(z_t, k_t) + \theta_{11} g_2(z_t, k_t) + \theta_{12} z_{1,t} + \theta_{14} z_{2,t})}{\lambda_t \left[1 + \phi \dfrac{g_1(z_t, k_t) - k_{1,t}}{k_{1,t}}\right]}\right] g_1(z_t, k_t) \tag{18}$$

$$\hat{g_2}(z_t, k_t) = \left[\frac{\beta(\theta_7 + 2\theta_9 g_2(z_t, k_t) + \theta_{11} g_1(z_t, k_t) + \theta_{13} z_{1,t} + \theta_{15} z_{2,t})}{\lambda_t \left[1 + \phi \dfrac{g_2(z_t, k_t) - k_{2,t}}{k_{2,t}}\right]}\right] g_2(z_t, k_t) \tag{19}$$

These updating equations use the fact that, as the arbitrary policy converges to the optimum, the term inside the square brackets converges to one and vice-versa.

[4]This representation was originally used in the context of parameterized expectations approach in Den Haan (1990), Marcet and Lorenzoni (1998) and subsequently in Judd et al. (2011). We need not worry about the $t$-measurability of period $t+1$ variables as the first order conditions do not contain expectational terms.

| Parameters | Values | Meaning |
|---|---|---|
| $\alpha$ | 0.36 | Curvature of the production function |
| $\phi$ | 0.5 | Parameter of cost function |
| $A$ | 0.028 | Parameter of production technology |
| $\sigma$ | 1 | Coefficient of relative risk aversion |
| $\beta$ | 0.99 | Discount rate |
| $\rho$ | 0.95 | Persistence of productivity process |
| $\sigma_\epsilon$ | 0.01 | Standard deviation of shocks to productivity |
| $k_{ss}$ | 1 | Steady-state capital |
| $T$ | 1000 | Length of each simulation |
| $\eta_v$ | 1 | Learning rate of value |
| $\eta_g$ | 0.01 | Learning rate of policy |

Table 4: Multi-country RBC Model - Parameters

| N | Dimension | Run Time(secs) | Max. Error | Avg. Error |
|---|---|---|---|---|
| 2 | 4 | 523 | 3.3e-04 | 7.6e-05 |
| 4 | 8 | 781 | 4.47e-04 | 1.02e-04 |
| 6 | 12 | 1138 | 5.18e-04 | 1.08e-04 |

Table 5: Multi-country RBC Model - Performance

- Generate $\{\hat{g}(z_t, k_t)\}_{t=1}^{T}$ using equations (18) and (19).
- Regress $\{\hat{g}(z_t, k_t)\}_{t=1}^{T}$ on $\{z_t, k_t\}_{t=1}^{T}$ to obtain the new policy coefficients $\hat{\theta}_g$
- $\theta_g = (1 - \eta_g)\theta_g + \eta_g\hat{\theta}_g$

Since we do not explicitly solve the nonlinear equations using a solver, policy improvement step can be extended in a straightforward way to a model with more than just two countries.

Table 4 gives the parameter values used to solve the multi-country model. All the parameter values of the model are taken from Juillard and Villemot (2011). The number of periods in each iteration is maintained at 1000. The learning rate of the policy($\eta_g$) is set to be very low at 0.01. Even a little higher value of $\eta_g$ leads to an immediate divergence of the algorithm.

Table 5 gives the computational time and errors associated with the solution to the multi-country model. $N$, as before, refers to the number of countries while dimension refers to the dimensionality of the state space(which is $2N$). As can be seen, this algorithm performs incredibly well both in terms of computational time and accuracy when we introduce more countries . The runtime increases concavely with no big increase in error statistics as the dimensionality of the problem increases. This algorithm shows great promise in solving elaborate dynamic economic models with a large state space.

### 4.3 Lumpy Investment Model

The next application of our algorithm is the lumpy investment model of Khan and Thomas (2003). This is an equilibrium model having heterogeneous production units distributed over their capital holdings and a representative household who owns all the plants. The production units in this model face a discrete choice of capital adjustment in a business cycle framework.

Each production unit has an associated capital stock $k$ and a fixed cost $\xi \in [0, B]$ which is drawn from a stationary distribution $G$. The fixed cost is denominated in terms of hours of labor. The output of the plant is given by $y = zk^{\theta}n^{\nu}$ where $z$ is the aggregate level of productivity and the production function is common across all the production units. After its production, each plant has to decide whether to pay its fixed cost realization and adjust its capital stock or make no changes and let its capital holdings depreciate. Since the fixed cost draw differs across the plants, the adjustment decision of each plant yields a distribution of plants (denoted by $\mu$) over the capital.

The aggregate states of the economy are $\{z, \mu\}$ while the individual states of each plant are $\{k, \xi\}$. The law of motion of the distribution is a function of the aggregate states and is given by $\mu' = \Gamma(z, \mu)$ where $\mu'$ refers to the next period's distribution. Plant's problem, after imposing the household's equilibrium conditions is represented in units of marginal utility of consumption($p(z, \mu)$) as follows. We do not go into further details of the model setup. The original article contains an in-depth explanation of formulating the plant's problem.

$$
\begin{aligned}
v^1(k, \xi; z, \mu) &= \max_n (zk^{\theta}n^{\nu} - \omega n + (1-\delta)k)p + \\
\max \Bigg\{ -\xi\omega p &+ \max_{k'}(-\gamma k' p + \beta \int_{z'} v^0(k'; z', \mu') H(dz'|z)), \\
&-(1-\delta)kp + \beta \int_{z'} v^0 \left( \frac{1-\delta}{\gamma} k, ; z', \mu' \right) H(dz'|z) \Bigg\}
\end{aligned}
\tag{20}
$$

where

$$
v^0(k; z, \mu) = \int_0^B v^1(k, \xi; z, \mu) G(d\xi) \tag{21}
$$

Before proceeding to our solution algorithm, let us give a concise summary of the results used in the original algorithm. The value of undertaking capital adjustment is given by:

$$E(z,\mu) = \max_{k'} \left( -\gamma k' p + \beta \int_{z'} v^0(k'; z', \mu') H(dz'|z) \right) \tag{22}$$

Let the common level of capital that maximizes the above optimization be $k^*(z,\mu)$. From the original functional equation (20), each plant faces a cutoff value of the adjustment cost($\bar{\xi}$) and the plant will adjust if the cost draw is below the cutoff value and will not adjust if the cost is above the cutoff value.

$$\bar{\xi}(k,z,\mu) = min\{B, max\{0, \hat{\xi}_k\}\}$$

where

$$-p(z,\mu)\hat{\xi}_k \omega(z,\mu) + E(z,\mu) = -p(z,\mu)(1-\delta)k + \beta \int_{z'} v^0 \left( \frac{1-\delta}{\gamma} k, ; z', \mu' \right) H(dz'|z) \tag{23}$$

Let $k^f(k,\xi;z,\mu)$ denote the capital stock that a plant, with capital stock $k$ and fixed cost draw $\xi$, will start the next period with.

$$k' = k^f(k,\xi;z,\mu) = \begin{cases} k^*(z,\mu) \text{ if } \xi \leq \bar{\xi}(k,z,\mu) \\ \frac{1-\delta}{\gamma} k \text{ if } \xi > \bar{\xi}(k,z,\mu) \end{cases} \tag{24}$$

Using equation (24), we can explicitly define the law of motion of the distribution, $\mu' = \Gamma(z,\mu)$ as follows. For $k \neq k^*(z,\mu)$,

$$\mu'(k) = \left[ 1 - G\left( \bar{\xi}\left( \frac{\gamma}{1-\delta} k; z, \mu \right) \right) \right] \mu \left( \frac{\gamma}{1-\delta} k \right) \tag{25}$$

and if $k = k^*(z,\mu)$

$$\mu'(k) = \int_K G(\bar{\xi}(k;z,\mu))\mu(dk) + \left[ 1 - G\left( \bar{\xi}\left( \frac{\gamma}{1-\delta} k; z, \mu \right) \right) \right] \mu \left( \frac{\gamma}{1-\delta} k \right) \tag{26}$$

where $K \subseteq \mathbb{R}_+$ is the set of capital levels. The market clearing quantities of consumption and hours are given by

$$C = \int_K \left( zF(k, n^f(k;z,\mu)) - G(\bar{\xi}(k;z,\mu))[\gamma k^*(z,\mu) - (1-\delta)k] \right) \mu(dk) \tag{27}$$

$$N = \int_K \left[ n^f(k;z,\mu) + \int_0^{\bar{\xi}(k;,z,\mu)} \xi G(d\xi) \right] \mu(dk) \tag{28}$$

where $n^f(k;z,\mu)$ is the hours of labor chosen by a plant with capital holdings $k$. The model is difficult to solve because the state space consists of the distribution($\mu$) which is a high dimensional object. Khan and Thomas (2003) use a generalization of Krusell et al. (1998) algorithm to solve their model. Khan and Thomas (2003) replace the high dimensional object with a collection of moments implied by the distribution to reduce the dimensionality of the problem. In the original Krusell et al. (1998) model, the prices are determined immediately once the aggregate state vector is known which is not true in the case of lumpy investment model. So, Khan and Thomas (2003) include an additional forecasting equation for the price along with the ones for the aggregate moments. Khan and Thomas (2003) show that a function of just the mean of the distribution and aggregate productivity is an excellent predictor of tomorrow's aggregate state. So, they approximate the state space by replacing distribution($\mu$) with its mean($m$) to solve the model[5]. The rules forecasting the current price $p$ and the next period's aggregate moment $m'$ are denoted by $p = \hat{p}(z, m; \chi_p)$ and $m' = \hat{\Gamma}(z, m; \chi_m)$ respectively.

Khan and Thomas (2003) solve for $v^0$ on a multidimensional grid of points and they use tensor product splines to interpolate the values inside the grid. The original two-step algorithm is as follows.

- *Inner Loop:* Using the current estimates of $\chi_p$ and $\chi_m$, solve for $v^0$ on a predefined grid of points on $z, k, m$ using equations (20)-(23).

- *Outer Loop:* This step simulates the economy over $T$ periods obtaining the distribution of plants $\mu_t$ at each time period. The aggregate moment $m$ is directly calculated from $\mu$, and using $m'$ from the assumed law of motion, the continuation value $\beta \int_{z'} v^0(k'; z', \mu') H(dz'|z))$ is determined for any $k'$. By imposing the market clearing conditions (27) and (28), the equilibrium price $p$ can be obtained at each period. Using equations (25) and (26), obtain $\mu_{t+1}$ to go to the next period of the simulation. At the end of the simulation, the data obtained on $\{p_t, m_t\}_{t=1}^{T}$ are used to update the estimates of $\chi_p$ and $\chi_m$.

In our implementation of the solution method, the first step of the original algorithm is replaced with the simulation routine as explained below while the second step(outer loop) is identical to what Khan and Thomas (2003) do. The two main differences of this solution method from the original are:

- The first step(inner loop) uses a simulation based algorithm to solve for $v^0$ instead of the

[5] Khan and Thomas (2003) also repeat the solution by replacing the distribution with two conditional means instead of an aggregate mean, with practically no change in the results.

grid based algorithm the original article uses.

- As a by-product of the simulation approach, the aggregate productivity $z$ is treated as a continuous variable compared to the discretized process used in the original method. Because of this, the forecast rules for both the price and next period's moment includes the aggregate productivity as an explicit regressor while the original article derives conditional forecast rules for different discrete values of $z$. Let the forecast rules for the current price($p$) and next period's aggregate moment($m'$) be given by $p = \hat{p}(z, m; \theta_p)$ and $m' = \hat{\Gamma}(z, m; \theta_m)$ respectively.

The simulation version of the algorithm is implemented as follows.

- Assume a functional form for $v^0$. Let the coefficient vector be $\theta_v$.
- Given the initial productivity realization $z_0$, simulate the productivity realizations $\{z_t\}_{t=1}^T$ for $T$ periods.
- While ($\theta_p$ or $\theta_m$ doesn't converge)
  - Given the initial aggregate moment $m_0$ and the forecast rules, simulate $\{p_t\}_{t=0}^T$ and $\{m_{t+1}\}_{t=0}^T$.
  - While ($\theta_v$ doesn't converge)
    - for $t = 1$ to $T$
      - Calculate $\{v_t^0\}_{t=1}^T$ using the current approximation $\theta_v$, $\{z_t\}_{t=1}^T$,$\{p_t\}_{t=1}^T$, $\{m_t\}_{t=1}^T$ and equations (20)-(23).
    - End for.
    - Regress $\{v_t^0\}_{t=1}^T$ on $\{z_t, p_t, m_t\}_{t=1}^T$ to obtain the updated coefficients $\hat{\theta_v}$.
    - $\theta_v = (1 - \eta_v)\theta_v + \eta_v\hat{\theta_v}$.
  - End while.
  - Execute the outer loop with the updated approximation for $v^0$ to obtain improved estimates for $\theta_m$ and $\theta_p$.
- End while.

| Parameters | Values | Meaning |
|---|---|---|
| $\gamma$ | 1.016 | Trend growth rate |
| $\beta$ | 0.954 | Discount rate |
| $\delta$ | 0.06 | Depreciation rate |
| $\theta$ | 0.325 | Capital share of output |
| $\nu$ | 0.580 | Labor share of output |
| $A$ | 3.614 | Marginal utility of leisure |
| $\rho$ | 0.9225 | Persistence of productivity process |
| $\sigma_\epsilon$ | 0.0134 | Standard deviation of shocks to productivity |
| $T$ | 1000 | Length of simulation in each iteration |
| $\eta_v$ | 1 | Learning rate. |

Table 6: Lumpy Investment Model - Parameters

| Regression | $\beta_1$ | $\beta_2$ | $\beta_3$ | $SE$ | $R^2$ |
|---|---|---|---|---|---|
| $m'$ | 0.0059 | 0.3228 | 0.8488 | 8.71e-6 | 0.9979 |
| $p$ | 1.1587 | -0.6994 | -0.4469 | 1.25e-5 | 0.9952 |

Each regression is of the form $logy = \beta_1 + \beta_2 logz + \beta_3 logm$.

Table 7: Lumpy Investment Model - Forecasting rules

Table 6 lists the parameters used to solve the model. These parameters are identical to the ones used by Khan and Thomas (2003) to solve the model. The inner loop iteration is very stable and hence can sustain a learning rate of 1. But a learning rate of 0.3 had to be used for the update of $\theta_m$ and $\theta_p$ in the outer loop else the algorithm oscillates and fails to converge. The algorithm took 32 minutes to converge which can be improved further upon careful calibration of algorithmic parameters.

Table 7 gives the converged forecast rules obtained from the simulation algorithm. The standard errors and $R^2$ associated with both the regressions reinforce the earlier finding that statistical mean alone(along with the aggregate productivity) is a very good predictor of the aggregate states. The standard errors of both the regressions are lower than their counterparts in the original article.

Table 8 compares the business cycle statistics of a representative firm model, which is the

| Algorithm | Y | I | C | N | w |
|---|---|---|---|---|---|
| *Standard deviations* | | | | | |
| Khan-Thomas(2003) | 1.91 | 6.37 | 0.94 | 1.10 | 0.94 |
| Simulation | 1.87 | 6.39 | 0.91 | 1.11 | 0.91 |
| *Contemporaneous correlations with output* | | | | | |
| Khan-Thomas(2003) | 1.0000 | 0.971 | 0.924 | 0.946 | 0.924 |
| Simulation | 1.0000 | 0.967 | 0.909 | 0.942 | 0.909 |

*log data HP filtered with $\lambda$ of 100. Y = output, I = investment, C = consumption, N = employment, w = wage.*

Table 8: Representative Firm Model - Business Cycle Moments

| Algorithm | Y | I | C | N | w |
|---|---|---|---|---|---|
| *Standard deviations* | | | | | |
| Khan-Thomas(2003) | 1.91 | 6.37 | 0.93 | 1.10 | 0.93 |
| Simulation | 1.76 | 5.49 | 0.92 | 0.92 | 0.92 |
| *Contemporaneous correlations with output* | | | | | |
| Khan-Thomas(2003) | 1.0000 | 0.972 | 0.926 | 0.947 | 0.926 |
| Simulation | 1.0000 | 0.975 | 0.954 | 0.955 | 0.955 |

*log data HP filtered with $\lambda$ of 100. Y = output, I = investment, C = consumption, N = employment, w = wage.*

Table 9: Lumpy Investment Model - Business Cycle Moments

lumpy investment model with no fixed costs for capital adjustment. The statistics implied by both the algorithms are very close to each other in this case. Table 9 gives the business cycle moments of the lumpy investment model implied by the algorithms. In the case of simulation based algorithm, labor, investment (and hence output)is less volatile compared to the benchmark algorithm. Consumption (and hence wages) are more strongly correlated with output compared to their counterparts in the benchmark case. This needs to be further investigated to find out the reasons for the observed difference between the statistics.

One of the important ways forward is to attempt at relaxing the state space approximation used in this model and other models of this type. The previous section showed that simulation based algorithm is scalable to high dimensional applications. Hence it is a good candidate for accomplishing this even though approximating the law of motion of the distribution would be a challenge. Jirnyi and Lepetyuk (2011) attempts to relax the state space approximation in the incomplete markets model of Krusell et al. (1998).

### 4.4 Sovereign Debt and Default Model

The final application of the simulation based algorithm is the sovereign default model of Arellano (2008). This paper considers a small open economy receiving a stochastic endowment($y$) each period. The main objective of the government is to maximize the expected lifetime utility of the representative agent. Each period, the sovereign has 2 decisions to make. First, it decides whether to default on its existing debt($b$) or not. If it decides to honor the existing debt, then the government can choose its next period's debt or savings($b'$) by issuing one period non-contingent bonds priced competitively. If the country chooses to default, the current outstanding debt is written off, but the country faces two penalties. It has to suffer an instantaneous penalty in terms of lost output and the country is banished to stay in autarky for this period. With a probability $\theta$ the sovereign regains access to the credit market next period, and with probability $(1-\theta)$, the country remains in autarky.

Let $v^c$ denote the value of honoring the contract(repay the debt) and $v^d$ the value of default.

$$v(b,y) = \max_{\{c,d\}}\{v^c(b,y), v^d(y)\} \tag{29}$$

$$v^d(y) = u(y^{def}) + \beta \int_{y'} [\theta v(0, y') + (1-\theta) v^d(y')] f(y', y) dy' \tag{30}$$

$$v^c(b, y) = \max_{b'} \left[ u(y - q(b', y)b' + b) + \beta \int_{y'} v(b', y') f(y', y) dy' \right] \tag{31}$$

where $q(b', y)$ represent the bond price at which the foreign lenders are willing to lend $b'$ to the country with a current endowment of $y$. Lenders borrow or lend as long as the expected return on the bonds equal the risk free rate $r$ (risk neutral). In equilibrium, the bond price is

$$q(b', y) = \frac{1 - \delta(b', y)}{1 + r} \tag{32}$$

$\delta(b', y)$ is the probability that the country will default on its debt $b'$ next period.

The utility function is assumed to be the standard CRRA utility and the endowment follows a log-normal AR(1) process. $y^{def}$ is the penalized output associated with a default episode and is given by

$$y^{def} = \begin{cases} \hat{y} \ \ if \ y \geq \hat{y} \\ \\ y \ \ if \ y < \hat{y} \end{cases} \tag{33}$$

The default models are difficult to solve as the value function $v$ is no longer smooth. So, discrete state space technique is the most common method to solve these models. Even though discrete state space techniques are flexible and hence can solve non-smooth value functions, they also suffer from the curse of dimensionality. Also, Hatchondo et al. (2010) show that discrete state space method might lead to large approximation errors unless a significantly large number of grid points are used. In this exercise, we attempt to apply the simulation algorithm to the default model. Since this algorithm treats the state variables as continuous, this could be one way to handle the criticism of Hatchondo et al. (2010).

As before, the algorithm that we use to solve the model uses stochastic simulation to generate data points and iterates on the equations (29)-(31).

- Assume a functional form for $v^c$ and $v^d$. Let the coefficient vector be $\gamma_c$ and $\gamma_d$ respectively.
- Given the initial output realization $y_0$, simulate the outputs $\{y_t\}_{t=1}^T$.
- while ($\gamma_c$ or $\gamma_d$ doesn't converge)
  - for $t = 1$ to $T$
    - If the sovereign is in autarky at period $t$, grant access to credit market with probability $\theta$.

- 
    - * Calculate $v_t^d$ using equation (30).
      * If the country has access to credit market, find the maximizing choice of $b'$ and $v^c$ using equation (31) and (32).
      * If the sovereign defaults at period $t$, exclude the country from credit market at $t+1$, else $b_{t+1} = b'$.
    - End for.
    - Using $\{v_t^c\}_{t=1}^T$, $\{v_t^d\}_{t=1}^T$, $\{y_t\}_{t=1}^T$ and $\{b_t\}_{t=1}^T$, update the approximation $\gamma_c$ and $\gamma_d$.
- End while.

In the above algorithm, in order to calculate $b'$ and in turn $v^c$, we need to calculate the probability of default $\delta(b', y)$. This is done as follows.

- If $b' \geq 0$ (i.e. the country saves)
    - Default probability($\delta(b', y)$) is 0.
- else
    - Find the threshold level of tomorrow's output($\bar{y}$), below which the country will default on the debt($b'$) tomorrow.
    - Probability of default ($\delta(b', y)$) is $F(\bar{y})$, where $F$ is the CDF of y.
- End if.

To perform the maximization in (31), we start with an initial candidate($b_0$) by finding the maximum on a discretized grid on the space of $b$. Using that as an initial guess, we use local optimization routines around $b_0$ to find the optimal debt $b'$. This procedure follows closely with the one used by Hatchondo et al. (2010).

In this implementation, the value functions $v^c$ and $v^d$ are parameterized using quadratic polynomials as follows.

$$\begin{aligned} v^c(b, y) &= \alpha_0 + \alpha_1 b + \alpha_2 y + \alpha_3 b^2 + \alpha_4 y^2 + \alpha_5 by \\ v^d(y) &= \theta_0 + \theta_1 y + \theta_2 y^2 + \theta_3 D + \theta_4 Dy + \theta_5 Dy^2 \end{aligned}$$

where

$$D = \begin{cases} 1 \; if \; y \geq \hat{y} \\ 0 \; if \; y < \hat{y} \end{cases}$$

| Parameters | Values | Meaning |
|---|---|---|
| $r$ | 0.017 | Risk-free rate |
| $\sigma$ | 2 | Coefficient of relative risk aversion |
| $\beta$ | 0.953 | Discount rate |
| $\theta$ | 0.282 | Probability of redemption |
| $\hat{y}$ | 0.9718 | Cutoff for default penalty |
| $\rho$ | 0.945 | Persistence of productivity process |
| $\sigma_\epsilon$ | 0.025 | Standard deviation of shocks to productivity |

Table 10: Sovereign Default Model - Parameters

The functional form for $v^d$ has a dummy variable to factor-in the kink (at $\hat{y}$) introduced by the penalty scheme. We use least squares regression with Tikhonov's regularization for updating the functional approximation.

Table 10 lists the parameter values used to solve this model. These are the values used in the original Arellano (2008).

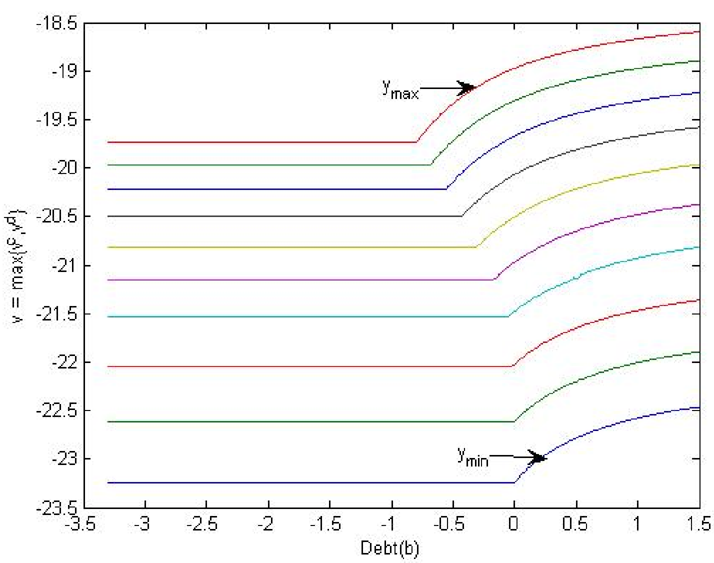


Figure 1: Sovereign Default Model - Value Function

Figure 1 shows the value function obtained from the algorithm. The algorithm, at the time of writing, is not stable and the coefficients oscillate without achieving a monotone convergence like previous applications. Even though the value function obtained thus far shows promise, more work needs to be done in order to stabilize the algorithm for this application.

Using stochastic learning rate schedules, as documented in George and Powell (2006) might help in increasing the stability and the speed of convergence of the algorithm. Another promising way forward in using stochastic simulations to solve models with discrete choice is to incorporate the envelope theorem developed in Clausen and Strub (2012).

# 5 Conclusion and Future Work

In this paper, I develop an algorithm based on stochastic simulation framework and demonstrate its usefulness in high dimensional applications by solving a multi-country business cycle model. I also use this algorithm to solve the lumpy investment model of Khan and Thomas (2003). Finally, I apply the algorithm to the sovereign default model of Arellano (2008) and more work needs to be done on this front. Implementing stochastic simulation based algorithms for solving models with discrete choice would be a valuable addition to the existing set of tools. Another useful direction for future research is to leverage on these algorithms to relax the state space approximation used in the solution methods of heterogeneous agent economies.

## References


Arellano, C. (2008). Default risk and income fluctuations in emerging economies. *American Economic Review*, 98(3):690–712.

Arellano, C., Maliar, L., Maliar, S., and Tsyrennikov, V. (2014). Envelope Condition Method with an Application to Default Risk Models. BYU Macroeconomics and Computational Laboratory Working Paper Series 2014-04, Brigham Young University, Department of Economics, BYU Macroeconomics and Computational Laboratory.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, 1st edition.

Buşoniu, L., Babuška, R., De Schutter, B., and Ernst, D. (2010). *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, Boca Raton, Florida.

Clausen, A. and Strub, C. (2012). Envelope theorems for non-smooth and non-concave optimization. ECON - Working Papers 062, Department of Economics - University of Zurich.

Den Haan, W. J. (1990). The optimal inflation path in a Sidrauski-type model with uncertainty. *Journal of Monetary Economics*, 25(3):389–409.

Den Haan, W. J. and Marcet, A. (1990). Solving the stochastic growth model by parameterizing expectations. *Journal of Business & Economic Statistics*, 8(1):31–34.

George, A. and Powell, W. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1):167–198.

Haan, W. J. D., Judd, K. L., and Juillard, M. (2011). Computational suite of models with heterogeneous agents ii: Multi-country real business cycle models. *Journal of Economic Dynamics and Control*, 35(2):175 – 177. Computational Suite of Models with Heterogeneous Agents II: Multi-Country Real Business Cycle Models.

Hatchondo, J. C., Martinez, L., and Sapriza, H. (2010). Quantitative properties of sovereign default models: solution methods. *Review of Economic Dynamics*, 13(4):919–933.

Hull, I. (2015). Approximate dynamic programming with post-decision states as a solution method for dynamic economic models. *Journal of Economic Dynamics and Control*, 55(0):57 – 70.

Jirnyi, A. and Lepetyuk, V. (2011). A reinforcement learning approach to solving incomplete market models with aggregate uncertainty. Working Papers. Serie AD 2011-21, Instituto Valenciano de Investigaciones Econmicas, S.A. (Ivie).

Judd, K. L., Maliar, L., and Maliar, S. (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. *Quantitative Economics*, 2(2):173–210.

Juillard, M. and Villemot, S. (2011). Multi-country real business cycle models: Accuracy tests and test bench. *Journal of Economic Dynamics and Control*, 35(2):178 – 185. Computational Suite of Models with Heterogeneous Agents II: Multi-Country Real Business Cycle Models.

Khan, A. and Thomas, J. K. (2003). Nonconvex factor adjustments in equilibrium business cycle models: do nonlinearities matter? *Journal of Monetary Economics*, 50(2):331 – 360.

Krusell, P., Smith, A. A., and Jr. (1998). Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy*, 106(5):867–896.

Marcet, A. (1991). Solving non-linear stochastic models by parameterizing expectations: An application to asset pricing with production. Economics Working Papers 5, Department of Economics and Business, Universitat Pompeu Fabra.

Marcet, A. and Lorenzoni, G. (1998). Parameterized expectations approach; Some practical issues. Economics Working Papers 296, Department of Economics and Business, Universitat Pompeu Fabra.

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons.

Tauchen, G. (1986). Finite state markov-chain approximations to univariate and vector autoregressions. *Economics letters*, 20(2):177–181.

Van Roy, B., Bertsekas, D. P., Lee, Y., and Tsitsiklis, J. N. (1997). A neuro-dynamic programming approach to retailer inventory management. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 4, pages 4052–4057. IEEE.