The Ronald O. Perelman Center for Political Science and Economics (PCPSE)
133 South 36th Street
Philadelphia, PA 19104-6297

pier@econ.upenn.edu
http://economics.sas.upenn.edu/pier

PIER Working Paper
20-028

FiPIt: A Simple, Fast Global Method for Solving Models with Two Endogenous States & Occasionally Binding Constraints

ENRIQUE G. MENDOZA
University of Pennsylvania, NBER & PIER

SERGIO VILLALVAZO
University of Pennsylvania

July 20, 2020

https://ssrn.com/abstract=3657211

FiPIt: A Simple, Fast Global Method for Solving Models with Two Endogenous States& Occasionally Binding Constraints*

Enrique G. Mendoza
University of Pennsylvania, NBER & PIER

Sergio Villalvazo
University of Pennsylvania

Abstract

We propose a simple and fast fixed-point iteration algorithm (*FiPIt*) to obtain the global, non-linear solution of macro models with two endogenous state variables and occasionally binding constraints. This method uses fixed-point iteration on Euler equations to avoid solving two simultaneous non-linear equations (as with the time iteration method) or creating modified state variables requiring irregular interpolation (as with the endogenous grids method). In the small-open-economy RBC and Sudden Stops models provided as examples, *FiPIt* is much faster than time iteration and various hybrid methods.

*We thank Javier Bianchi, Pablo D'Erasmo, Bora Durdu, Vincenzo Quadrini and Urban Jermann for helpful comments and suggestions. Contact email addresses: egme@sas.upenn.edu and vsergio@sas.upenn.edu.

1 Introduction

Important branches of the recent macroeconomics literature study quantitative solutions of models in which constraints are triggered endogenously (i.e. they are "occasionally binding"), as in studies of the zero-lower-bound on interest rates or financial crises triggered by credit constraints. Because these models typically feature non-linear decision rules that lack analytic solutions and capture precautionary savings, global solution methods (e.g. time iteration or endogenous grids methods) are the preferable tool for solving them. Global methods are, however, less practical than perturbation methods, because of limitations that make them slow and difficult to implement with widely used software (e.g. Matlab). On the other hand, perturbation methods for solving models with occasionally binding constraints, such as OccBin developed by Guerrieri and Iacoviello [2015] and DynareOBC proposed by Holden [2016], have caveats that limit the scope of the findings that can be derived from using them (see Aruoba et al. [2019], Durdu et al. [2019])

This paper proposes a simple and fast algorithm to obtain the global solution of models with two endogenous states and occasionally binding constraints. This algorithm is denoted as *FiPIt* because it is based on the well-known fixed-point iteration approach to solve systems of transcendental equations. It is easy to implement in a Matlab platform and is significantly faster than the standard time iteration algorithm and several hybrid alternatives. *FiPIt*'s solution strategy builds on the class of time iteration methods that originated in the work of Coleman [1990], who first proposed a global solution method based on policy function iterations of the Euler equation. Since then, various enrichments and modifications of this approach have been developed, in particular the endogenous grids method proposed by Carroll [2006] (see Rendahl [2015] for a general discussion of these methods and an analysis of their convergence properties). *FiPIt* differs from these methods in that it applies the fixed-point iteration method to solve a model's Euler equations. For instance, in the Sudden Stops model solutions provided as example in this paper, the bonds (capital) Euler equation is used to solve directly for a "new" bonds decision rule (capital pricing function) without the need of a non-linear solver. The capital decision rule is solved for in "exact" form using the models' optimality conditions.

The endogenous grids method also avoids using a non-linear solver, but it does so by defining alternative state variables so that obtaining analytic solutions of Euler equations for control variables (e.g. consumption, investment) requires irregular interpolation of functions defined over endogenous grids of the original state space. This is innocuous in one-dimensional problems, but in two- and higher-dimensional problems it requires elaborate interpolation methods to tackle the non-rectangular nature of the endogenous grids. In particular, Ludwig and Schön [2018] developed a method using Delaunay interpolation, and showed that it is significantly faster that standard time iteration.[1] Alternatively, Brumm and Grill [2014] proposed a a variant of the time iteration method that still uses a non-linear solver but gains speed and accuracy by updating grid nodes to track decision rule kinks using also Delaunay interpolation. In contrast, *FiPIt* retains the original state variables so that standard multi-linear interpolation on regular grids can be used.

We apply the algorithm to solve the model proposed by Mendoza [2010], which is a model of Sudden Stops (financial crises) in a small open economy. This model includes an occasionally binding credit constraint limiting intertemporal debt and working capital not to exceed a fraction of the market value of physical capital (i.e. pledgeable collateral). The results show that, relative to the time iteration method, *FiPIt* reduces execution time by a factor of 2.5 (or 18.1 when solving an RBC variant of the model).[2] We also found that *FiPIt* continues to perform well for several parameter variations, despite the well-known drawback of fixed-point iteration methods indicating that their convergence is not guaranteed. Execution times for seven parameter variations of the model were smaller than using time iteration by factors of 2.0 to 18.1. Ludwig and Schön [2018] report reductions by factors of 2.7 to 4.1 using endogenous grids with Delaunay interpolation v. standard time iteration, or 1.8 to 2.5 using their hybrid method v. standard time iteration, when solving a perfect-foresight model of human capital accumulation in a small open economy.[3]

In addition to the Delaunay interpolation, a second drawback of the endogenous grids method relative to the *FiPIt* method is that it still requires a root-finder in order to determine equilibrium solutions in points of the state space in which occasionally binding constraints bind (see Ludwig and Schön [2018]). *FiPIt* requires a non-linear solver only if the solution of the allocations when the constraint binds cannot be separated from the solution of the multiplier of the constraint. The two are separable in models that feature several widely-used occasionally binding constraints, including

[1] Adjacent points in the endogenous grids do not generally match adjacent nodes in the matrix formed by the original grids. Ludwig and Schon tackled this problem using Delaunay interpolation. They also proposed a hybrid method that uses an exogenous grid for one of the endogenous states and an endogenous grid for the second.

[2] We used Matlab version R2017a on a Windows 10 laptop with an Intel Core i7-6700HQ 2.60GHz chip, 4 physical cores and 16 GB of RAM. The state space for the Sudden Stops (RBC) model has 72 (80) nodes on foreign assets and 30 on domestic capital. The Sudden Stops (RBC) model solved in 810 (100) seconds, compared with 1,986 (1,808) using the time iteration method.

[3] They report faster solution times for each individual scenario than with our algorithm but these are not comparable due to differences in models and hardware. We solve a stochastic model with three shocks, capital accumulation and adjustment costs, and a credit constraint that depends on the model's two endogenous states and a market price. They solve a deterministic model in which human capital is an accumulable factor produced with an exponential technology and a no-borrowing constraint. We do not have details about the software and hardware they used.

standard no-borrowing constraints, maximum debt limits, and constraints on debt-to-income and loan-to-value ratios that depend on endogenous variables. Solving variations of the SS model using these constraints, *FiPIt* reduced execution time relative to the time iteration method by a factor of 13.0 for a loan-to-value-ratio constraint and 17.9 for a maximum debt limit.

There are applications in the literature that solve models using fixed-point iteration algorithms with some features similar to the one we proposed here. Carroll [2011] described and implemented a fixed-point iteration algorithm for solving the workhorse complete-markets RBC model of a closed economy. Boz and Mendoza [2014], Bianchi and Mendoza [2018] and Bianchi et al. [2016] solved open-economy models with occasionally binding collateral constraints iterating on bond decision rules and/or pricing functions. All these applications considered only one endogenous state variable. Perri and Quadrini [2018] solved a two-country model with two endogenous state variables and a credit constraint resulting from an enforcement friction using Fortran and a state space with 121 points (11 nodes for each state variable). This paper differs from these studies in that we develop an algorithm that solves models with two endogenous states easily and fast in a standard Matlab platform and with a sizable state space including 2,160 points. *FiPIt* can be used in a variety of models with two endogenous states. The choice of functions that are iterated on using the Euler equations can vary across models, and there can be more that one arrangement for the same model.

The rest of the paper proceeds as follows. The next Section describes the principles of the algorithm in the simple case of a model of savings with endowment income, and uses this example also to explain how *FiPIt* differs from the time iteration and endogenous grids methods. Section 3 describes the Sudden Stops model and provides a step-by-step description of the complete algorithm. Section 4 provides quantitative results, evaluates the robustness of the algorithm, and conducts performance comparisons with alternative algorithms, including the standard time iteration method. Section 5 presents conclusions. In addition, the Matlab codes and an Appendix that provides a user's guide to the codes are available online.

2 A Fixed-Point Iteration Algorithm for a Simple Savings Model

We describe the principles of the *FiPIt* method using a savings model with stochastic endowment income and an exogenous interest rate. This model is a workhorse of various branches of the macro literature, including consumption and savings in partial equilibrium, heterogeneous agents models with incomplete markets, and international macro models of the small open economy.

A representative agent chooses consumption and savings plans so as to maximize a standard expected utility function:

$$E_0 \left\{ \sum_{t=0}^{\infty} \beta^t u(c_t) \right\}. \tag{1}$$

subject to the budget or resource constraint:

$$c_t = e^{z_t} \bar{y} + b_t - q b_{t+1}. \tag{2}$$

and a debt limit:

$$b_{t+1} \geq -\varphi. \tag{3}$$

In the utility function, $\beta \in (0, 1)$ is the subjective discount factor and $u(\cdot)$ is the period utility function, which can be any standard twice, continuously differentiable and concave utility function, although the CRRA functional form is the one used most often:

$$u(c_t) = \frac{c_t^{1-\sigma}}{1-\sigma},$$

where σ is the relative risk aversion coefficient. In the resource constraint, $e^{z_t}\bar{y}$ is stochastic income with mean $\bar{y}$ and shocks z_t of exponential support e^{z_t}, b_t are holdings of one-period, non-state-contingent discount bonds traded in a frictionless credit market. In a partial equilibrium model of savings or a model of a small open economy, the real interest rate r is exogenous, so the price of bonds is also exogenous and given by $q \equiv \frac{1}{1+r}$. In a general equilibrium model of heterogeneous agents, the above optimization problem is solved by each individual agent facing idiosyncratic income uncertainty, and the interest rate is endogenously determined so as to clear the bond market. The *FiPIt* method can be used in all of these models, except that in the heterogeneous agents model we would also need to iterate on the interest rate until the bond market clears. We focus on the small open economy case to simplify the exposition.

If the utility function satisfies the Inada condition and income shocks follow a discrete Markov process or a truncated continuous distribution, the debt limit follows from Aiyagari's Natural Debt Limit: agent's never choose optimal plans that leave them exposed to the risk of non-positive consumption, and hence never borrow more than the annuity value of the lowest income realization. Alternatively, agents may face an ad-hoc debt limit tighter than the natural debt limit. Thus, the model includes an occasionally binding constraint, albeit of a simple form: $b_{t+1} \geq -\varphi$.

The Euler equation for bond holdings is

$$u_c\left(c_t\right) = (1+r)\beta E_t\left[u_c(c_{t+1})\right] + \mu_t, \tag{4}$$

where $u_c(c_t)$ is the marginal utility of c_t and μ_t is the multiplier on the debt limit. Note that using the resource constraint to substitute for consumption, the Euler equation can be expressed as:

$$u_c\left(e^{z_t}\bar{y} + b_t - qb_{t+1}\right) = (1+r)\beta E_t\left[u_c\left(e^{z_{t+1}}\bar{y} + b_{t+1} - qb_{t+2}\right)\right] + \mu_t. \tag{5}$$

A competitive equilibrium for this economy is defined by stochastic sequences $[c_t, b_{t+1}]_{t=0}^{\infty}$ that satisfy equations (3) and (4) for all t. The economy has a well-defined limiting distribution of (b, y) (i.e. a stochastic steady state) only if $\beta(1+r) < 1$ (see Ljungqvist and Sargent [2012], Ch. 18). This condition is also a general equilibrium outcome in heterogeneous agents models, because otherwise all agents would want an infinite amount of bonds, which is inconsistent with market clearing in the market of risk-free bonds.

Since there are no inefficiencies affecting the small open economy (other than the incompleteness of asset markets), the competitive equilibrium can be represented as the solution to the following dynamic programming problem:

$$V(b, z) = \max_{c,b'}\left\{\frac{c^{1-\sigma}}{1-\sigma} + \beta\sum_{z'}\pi(z', z)V\left(b', z'\right)\right\}, \tag{6}$$

subject to

$$c = e^z\bar{y} + b - qb'$$

$$b' \geq -\varphi$$

The solution to the above Bellman equation is characterized by a decision rule $b'(b, z)$ and the associated value function $V(b, z)$, and the decision rule together with the Markov process of the shocks induce a joint ergodic (unconditional) distribution of bonds and income $\lambda(b, z)$.

"Euler equation" methods typically solve for $b'(b, z)$ over a discrete state space of (b, z) pairs using the recursive equilibrium conditions that follow from the first-order-conditions of the above Bellman equation:

$$c(b, z)^{-\sigma} \geq \beta R\sum_{z'}\pi(z', z)\left(c(b'(b, z), z')\right)^{-\sigma} \tag{7}$$

$$c(b,z) = e^z \bar{y} + b - qb'(b,z). \tag{8}$$

The recursive equilibrium of the model is then defined as the pair of decision rules $c(b,z), b'(b,z)$ that satisfy these two conditions.

The *FiPIt* method poses a conjecture of the decision rule $\hat{b}'_j(b,z)$ in iteration j, defined over the nodes of discrete grids for b and z. Intermediate values are then found by interpolation. The function $\hat{b}'_j(b,z)$ uses the resource constraint to generate its associated consumption function as $c_j(b,z) = e^z \bar{y} + b - q\hat{b}'_j(b,z)$. Using this consumption function, the above first-order conditions can be combined into an equation that solves for the "new" consumption function:

$$c_{j+1}(b,z) = \left\{ \beta R \sum_{z'} \pi(z',z) \left(c_j(\hat{b}'_j(b,z), z') \right)^{-\sigma} \right\}^{-\frac{1}{\sigma}} \tag{9}$$

In the right-hand-side of this Euler equation, we need the value of c_{t+1}, which is obtained by evaluating the consumption function at the t+1-values of the state variables: $c_j(\hat{b}'_j(b,z), z'))$. Since $\hat{b}'_j(b,z)$ is defined only on the nodes of the grid of bonds, this consumption function is interpolated over its first argument in order to determine $c_j(\hat{b}'_j(b,z), z')$ (i.e. the value of c_{t+1} implied by the conjectured consumption function). Once this is done, the Euler equation solves directly for a new consumption function $c_{j+1}(b,z)$ without a non-linear solver. Using the resource constraint, this new consumption function yields a new decision rule for bonds $b'_{j+1}(b,z)$, which is re-set to $b'_{j+1}(b,z) = -\varphi$ if $b'_{j+1}(b,z) \leq -\varphi$. Then the decision rule conjecture is updated to $\hat{b}'_{j+1}(b,z)$ as a convex combination of $\hat{b}'_j(b,z)$ and $b'_{j+1}(b,z)$: $\hat{b}'_{j+1}(b,z) = (1-\rho)\hat{b}'_j(b,z) + \rho b'_{j+1}(b,z)$. The process is repeated until $b'_{j+1}(b,z) = \hat{b}'_j(b,z)$ for all (b,z) in the grids, up to a convergence criterion.

Three points raised by Judd [1998] about fixed-point iteration algorithms like this one are worth recalling. First, using collocation methods instead of solving for a finite state space, the fixed-point iteration method can be represented in a form analogous to the Parameterized Expectations method, because the latter is a fixed-point iteration method that uses simulation and regression to construct conditional expectations. Second, using $0 < \rho < 1$ ($\rho > 1$) to set the decision rule of the next iteration is useful to address possible instability (slow convergence) of the algorithm. Third, a finite state space may be preferable to collocation methods to define the decision rules depending on whether we expect decision rules to be smooth or to have strong curvature. The latter can be particularly important in models with occasionally binding constraints that depend on endogenous variables, such as credit constraints that depend on collateral prices and yield U-shaped decision rules because of the Fisherian debt-deflation mechanism (see Bianchi and Mendoza [2018]). This will be the case in the model solved in the next Section.

Fixed-point iteration differs from the time iteration method because the latter applies the conjectured decision rule $\hat{b}'_j(b,z)$ only to substitute for the term b_{t+2} in the right-hand-side of the Euler equation (5), and then uses a non-linear solver to solve the resulting non-linear equation for the optimal choice of b_{t+1} as a function of (z_t, b_t). Hence, we can think of the fixed-point iteration method as a "proxy time iteration method" that substitutes for the b_{t+1} in the right-hand-side of the Euler equation with a proxy that is defined to be the conjectured decision rule, instead of treating that b_{t+1} term as endogenous.[4] Fixed-point iteration is also different from the endogenous grids method, because it does not redefine the endogenous state variable and instead solves the problem over the original rectilinear grids (b,z). Still, fixed-point iteration retains the main computational advantage of the endogenous grids method, which is that the Euler equation is reduced to an equation with an analytic solution for the decision rule, avoiding the need to use non-linear solvers.

3 The *FiPIt* Method for Two-Dimensional Models

This Section provides a detailed description of the steps that the *FiPIt* method follows to solve a model with two endogenous states and an occasionally binding constraint. The model pertains to a small open economy with two endogenous states, capital (k) and net foreign assets (b), and a credit constraint. If the constraint never binds, the algorithm solves a standard RBC model of a small open economy, and if it binds it solves a model with endogenous financial crises or Sudden Stops.

3.1 Model structure and equilibrium conditions

The model is the same as in Mendoza [2010], except that the preferences with endogenous discounting are replaced with standard time-separable expected utility with exogenous discounting at rate β. The economy is inhabited by a representative firm-household with preferences defined over stochastic sequences of consumption c_t and labor supply L_t, for $t = 0, ..., \infty$, given by:

$$E_0 \left[\sum_{t=0}^{\infty} \beta^t \frac{(c_t - \frac{L_t^\omega}{\omega})^{1-\sigma}}{1-\sigma} \right] \tag{10}$$

[4] From this perspective, it may seem as if the *FiPIt* method solves the "incorrect" Euler equation. Yet, as the paper shows, the solutions satisfy the same equilibrium conditions and are negligibly different from those obtained using standard time iteration. This is because *FiPIt* is essentially an application of the standard fixed-point iteration approach to solve transcendental equations.

The agent chooses sequences of consumption, labor, investment, and holdings of real, one-period international bonds, b_{t+1} (the agent borrows when $b_{t+1} < 0$), so as to maximize the above utility function subject to the following budget and collateral constraints:

$$c_t(1+\tau)+k_{t+1}-(1-\delta)k_t+\frac{a}{2}\frac{(k_{t+1}-k_t)^2}{k_t} = A_tF(k_t,L_t,v_t)-p_tv_t-\phi(R_t-1)(w_tL_t+p_tv_t)-q_t^bb_{t+1}+b_t, \tag{11}$$

$$q_t^bb_{t+1}-\phi R_t(w_tL_t+p_tv_t) \geq -\kappa q_tk_{t+1}. \tag{12}$$

The right-hand-side of the budget constraint is the sum of net profits from production and the resources generated by trading assets abroad. Net profits are equal to gross production minus the cost of imported inputs minus the servicing of foreign working capital loans for labor and imported inputs. Gross output is represented by a constant-returns-to-scale technology, $A_tF(k_t,L_t,v_t) = \exp(\epsilon_t^A)Ak_t^\gamma L_t^\alpha v_t^\eta$, that requires capital, k_t, labor and imported inputs, v_t, to produce a tradable good sold at a world-determined price (normalized to unity without loss of generality). TFP is subject to a random shock ϵ_t^A with exponential support around a mean of A. Working capital loans pay for a fraction ϕ of the cost of imported inputs and labor in advance of sales. These loans are obtained from foreign lenders at the beginning of each period and repaid at the end. Lenders charge the world gross real interest rate $R_t = R\exp(\epsilon_t^R)$ on these loans, where ϵ_t^R is an interest rate shock around a mean value R. Imported inputs are purchased at an exogenous relative price in terms of the world's numeraire $p_t = p\exp(\epsilon_t^P)$, where p is the mean price and ϵ_t^P is a shock to the world price of imported inputs (i.e., a terms-of-trade shock). The shocks ϵ_t^A, ϵ_t^R, and ϵ_t^P follow a joint first-order Markov process. The resources generated by trading assets abroad are given by $-q_t^bb_{t+1}+b_t$, where q_t is the price of the international bonds, which satisfies $q_t^b = R_t^{-1}$.

The left-hand-side of the budget constraint is the sum of consumption expenditures, investment and capital adjustment costs. Gross investment is $i_t = k_{t+1}-(1-\delta)k_t$ and gross investment inclusive of adjustment costs is $\tilde{i}_t = k_{t+1}-(1-\delta)k_t+\frac{a}{2}\frac{(k_{t+1}-k_t)^2}{k_t}$. Since government expenditures are not included in the model, we include a time-invariant consumption tax τ that is used to calibrate the model to match the average share of government expenditures in GDP in the data. This is done so that consumption and investment shares in the model can match their data counterparts. Since the tax is constant, it does not distort the savings-consumption margin. The tax does distort labor supply but this distortion is constant over time, since the tax itself is constant.

The credit constraint limits the total debt, which is equal to intertemporal debt plus working capital financing, not to exceed the fraction κ of the market value of the end-of-period capital stock.

This is a more complex constraint than borrowing constraints of the class $b_{t+1} \geq -\varphi$, widely used in heterogeneous agents models and also in the algorithm proposed by Ludwig and Schön [2018]. Notice that the prices q_t and w_t that appear in this constraint (and the wage in the budget constraint), are endogenous market prices taken as given by the agent when solving its optimization problem. As in Mendoza [2010], the wage rate must be on the labor supply curve (i.e. it must equal the tax-adjusted marginal disutility of labor), which requires $w_t = L^{\omega-1}(1+\tau)$, and the price of capital must satisfy the optimality condition requiring that $q_t = \frac{\partial \tilde{i}_t}{\partial k_{t+1}}$. With these simplifications noted, the competitive equilibrium of the economy can be represented with the optimization problem of the firm-household, instead of defining separate problems for households and firms. This equilibrium, however, cannot be represented as the solution to a planner's problem formulated as a single Bellman equation, because the planner would internalize the responses of wages and asset prices to its optimal plans, while the representative firm-household does not.

Defining λ_t and μ_t as the future-value multipliers of the budget and collateral constraints respectively, the model's equilibrium conditions in sequential form are:

$$\left(c_t - \frac{L_t^{\omega}}{\omega}\right)^{-\sigma} = \lambda_t(1+\tau) \tag{13}$$

$$A_t F_{L_t}(k_t, L_t, v_t) = w_t\left(1 + \phi(R_t - 1) + \frac{\mu_t}{\lambda_t}\phi R_t\right) \tag{14}$$

$$A_t F_{v_t}(k_t, L_t, v_t) = p_t\left(1 + \phi(R_t - 1) + \frac{\mu_t}{\lambda_t}\phi R_t\right) \tag{15}$$

$$\lambda_t = \frac{1}{q_t^b}\beta E_t[\lambda_{t+1}] + \mu_t \tag{16}$$

$$\lambda_t = \frac{1}{q_t}\beta E_t\left[\lambda_{t+1}\left(\exp(\epsilon_{t+1}^A)F_{k_{t+1}} - \delta + \frac{a}{2}\frac{(k_{t+2}-k_{t+1})^2}{k_{t+1}^2} + q_{t+1}\right)\right] + \mu_t\kappa \tag{17}$$

$$q_t = \frac{\partial \tilde{i}_t}{\partial k_{t+1}} = 1 + a\left(\frac{k_{t+1}-k_t}{k_t}\right) \tag{18}$$

$$w_t = L^{\omega-1}(1+\tau) \tag{19}$$

$$c_t(1+\tau) + k_{t+1} - (1-\delta)k_t + \frac{a}{2}\frac{(k_{t+1}-k_t)^2}{k_t} = A_t F(k_t, L_t, v_t) - p_t v_t - \phi(R_t - 1)(L_t^{\omega}(1+\tau) + p_t v_t) - q_t^b b_{t+1} + b_t \tag{20}$$

Solving this model with the time iteration method requires solving the Euler equations (16) and (17) as part of a system of non-linear equations. Given conjectures of the decision rules for capital and bonds, and simplifying using the other equilibrium conditions, the two Euler equations form a two-equation system that yields the "new" decision rules. When the collateral constraint does not bind, these two Euler equations have their standard forms. When the constraint binds, the multiplier μ_t is an additional endogenous variable and there is an additional equation, which is the constraint holding with equality. The solution can still be reduced to a two-equation system, by using the constraint to substitute for $q_t^b b_{t+1}$ together with the conjectured decision rules so as to obtain a two-equation system in k_{t+1} and μ_t.[5]

Solving with the endogenous grid method requires defining grids for two alternative state variables (s^1, s^2) such that $s_t^1 \equiv q_t^b b_{t+1}$ and $s_t^2 \equiv k_{t+1}/(1-\delta)$, and then proceeding as in Ludwig and Schön [2018] to first determine the values of (b_{t+1}, k_{t+1}) associated with each (s_t^1, s_t^2) pair, then use the optimality conditions (including the Euler equations) to solve for the contemporaneous controls, particularly (c_t, i_t), and then use the resource constraint and the definition of gross investment to extract the implied values of the original endogenous states (b_t, k_t), namely the endogenous grids. When solving for the contemporaneous controls, the optimality conditions form a system of equations that has an analytic solution, thus avoiding the need to use a non-linear solver, but the endogenous grids of (b_{t+1}, k_{t+1}) are irregular, so interpolation of the relevant functions required to obtain the solution of the system is implemented using Delaunay interpolation.[6] As noted earlier, *FiPIt* does not need either non-linear solvers or interpolation methods for irregular grids.[7] Standard bi-linear interpolation over rectangular grids still applies.

3.2 Description of the *FiPIt* algorithm

The *FiPIt* method solves the model's equilibrium conditions in recursive form. The model has two endogenous states, b and k, and three exogenous states, using s to denote the triple of exogenous shocks $s \equiv (A, R, p)$, which includes shocks to TFP (A), the world interest rate (R) and the price of imported inputs (p). The recursive equilibrium is defined by a set of recursive functions for allocations $[b'(b,k,s), k'(b,k,s), c(b,k,s), L(b,k,s), v(b,k,s)]$, prices $[w(b,k,s), q(b,k,s)]$ and multipliers $[\lambda(b,k,s), \mu(b,k,s)]$ that satisfy the recursive representation of equations (13)-(20), which is provided in Section 2 of the Appendix.

[5] If the solution implies a value of b_{t+1} lower than the lower bound of the grid of bonds, we set b_{t+1} to that lower bound and solve again the two-equation system for the values of k_{t+1} and μ_{t+1} consistent with that value of b_{t+1}. Hence, the lower bound of the bonds grid is still treated as a constraint of the form $b_{t+1} \geq -\varphi$.

[6] The Ludwig-Schon algorithm still needs to solve a non-linear equation in order to solve for the contemporaneous controls in states in which their no-borrowing constraint binds.

[7] A non-linear equation may need to be solved for in states in which the credit constraint binds, depending on the structure of the constraint (as we explain in subsection 3.2), but this is separate from the need to solve a two-Euler-equation non-linear system when time iteration is used to solve models with two endogenous states.

The recursive equilibrium is solved for over a discrete state space, which requires defining discrete grids for (b, k, s). The grid for the shock triples $s \in \mathbf{S}$ comes from the discretization of the stochastic processes of the model's three shocks. This is typically done using Tauchen's quadrature method. Here we take $\mathbf{S}$ and the associated Markov transition probability matrix from Mendoza [2010], where $\mathbf{S}$ has eight triples (i.e. each shock has two realizations). For the endogenous states, we define grids with M nodes for bonds and N nodes for capital, respectively: $\mathbf{B} = \{b^1 < b^2 < ... < b^M\}$, $\mathbf{K} = \{k^1 < k^2 < ... < k^N\}$. The state space has $M \times N \times 8$ elements and is defined by all $(b, k, s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$. Once parameter values and the discrete state space are defined, the *FiPIt* algorithm is implemented following the steps described below.

Step 1. Start iteration j with conjectured functions for the price of capital $\hat{q}_j(b, k, s)$, the decision rule for bonds $\hat{b}'_j(b, k, s)$, and the multiplier ratio $\hat{\tilde{\mu}}_j(b, k, s) \equiv \mu_j(b, k, s)/\lambda_j(b, k, s)$. The first iteration can start with $\hat{\tilde{\mu}}_0(b, k, s) = 0$ so that the first pass runs as if it were an RBC model and only cases where the constraint binds pass positive multipliers to the next iteration. The initial functions can be set to $\hat{q}_0(b, k, s) = 1$ and $\hat{b}'_0(b, k, s) = b$, which imply stationary decision rules for capital and bonds. Note also that this same algorithm can be used to solve a standard RBC model without the occasionally binding constraint, by simply setting κ high enough so that the constraint never binds.

Step 2. Using the recursive equilibrium conditions, compute the iteration-j implied decision rules for capital $k'_j(b, k, s)$, consumption, investment (inclusive of adjustment costs), labor, inputs and output as shown below. Note that, given $\hat{q}_j(b, k, s)$, the capital decision rule has an analytic solution that follows from optimality condition (18) (i.e. the capital decision rule has a closed-form solution as a function of the price of capital). The factor allocation rules follow from the conditions equating marginal products with marginal costs, which include factor prices and financing costs. The wages bill wL is replaced with $(1 + \tau)L^\omega$ because of the optimality condition for labor supply. With these arguments in mind, the iteration-j implied decision rules are:

$$
\begin{aligned}
k'_j(b,k,s) =& \frac{k}{a}\left[\hat{q}_j(b,k,s) - 1 + a\right] && (21)\\
\tilde{i}_j(b,k,s) =& (k'_j(b,k,s) - k)\left[1 + \frac{a}{2}\left(\frac{k'_j(b,k,s) - k}{k}\right)\right] - \delta k && (22)\\
v_j(b,k,s) =& \left\{\frac{Ak^{\gamma}\eta^{\frac{\omega-\alpha}{\omega}}\frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}}\left[1+\phi(R-1)+\hat{\tilde{\mu}}_j(b,k,s)\phi R\right]}\right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}} && (23)\\
L_j(b,k,s) =& \left\{\frac{\alpha}{\eta(1+\tau)}pv_j((b,k,s)\right\}^{\frac{1}{\omega}} && (24)\\
y_j(b,k,s) =& Ak^{\gamma}L_j(b,k,s)^{\alpha}v_j(b,k,s)^{\eta} && (25)
\end{aligned}
$$

Consumption then follows from the resource constraint:

$$
\begin{aligned}
(1+\tau)c_j(b,k,s) = y_j(b,k,s) - pv_j(b,k,s) - \phi(R-1)\left[(1+\tau)L_j(b,k,s)^{\omega} + pv_j(b,k,s)\right]\\
- \tilde{i}_j(b,k,s) - \frac{\hat{b}'_j(b,k,s)}{R} + b
\end{aligned} \tag{26}
$$

Note that for points where $\hat{\tilde{\mu}}_j(b,k,s) = 0$, factor allocations and output are the same as for an RBC model without credit frictions, which because of the GHH structure of period utility (i.e. the marginal rate of substitution between c and L is independent of c) depend only on (k,s). We keep them as functions of all three states because when $\hat{\tilde{\mu}}_j(b,k,s) > 0$ factor allocations and output do depend on the three states.

Step 3. Assume the collateral constraint does not bind. This implies that the *new* decision rule for the modified multiplier is $\hat{\tilde{\mu}}_{j+1}(b,k,s) = 0$, and the *new* decision rules for the rest of the endogenous variables are solved using the recursive equilibrium conditions as follows:

3.1 Factor allocations and output again match the expressions corresponding to an RBC model with perfect credit markets:

$$
\begin{aligned}
v_{j+1}(b,k,s) =& \left\{\frac{Ak^{\gamma}\eta^{\frac{\omega-\alpha}{\omega}}\frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}}\left[1+\phi(R-1)\right]}\right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}} && (27)\\
L_{j+1}(b,k,s) =& \left\{\frac{\alpha}{\eta(1+\tau)}pv_{j+1}(b,k,s)\right\}^{\frac{1}{\omega}} && (28)\\
y_{j+1}(b,k,s) =& Ak^{\gamma}L_{j+1}(b,k,s)^{\alpha}v_{j+1}(b,k,s)^{\eta} && (29)
\end{aligned}
$$

3.2 Solve for c_{j+1} by applying the fixed-point iteration method to the Euler equation for bonds. The iteration-j conjectures for capital and bonds are used everywhere in the right-hand-side of this Euler equation, so that we obtain an analytic solution for c_{j+1}. Keep track of the subscripts denoting which function is used in each term:

$$\begin{aligned} c_{j+1}(b,k,s) \\ = \left\{ \beta R E \left[\left(c_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s') - \frac{L_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')^{\omega}}{\omega} \right)^{-\sigma} \right] \right\}^{-\frac{1}{\sigma}} \\ + \frac{L_{j+1}(b,k,s)^{\omega}}{\omega} \end{aligned} \tag{30}$$

In the above expression, the functions $c_j(b,k,s)$ and $L_j(b,k,s)$ are defined only at the nodes of $\mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$, but since the values of $\hat{b}'_j(b,k,s)$ and $k'_j(b,k,s)$ generally do not match node grids in $\mathbf{B}$ and $\mathbf{K}$, respectively, $c_j(\cdot)$ and $L_j(\cdot)$ are interpolated over their first two arguments to determine $c_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')$ and $L_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')$. Standard bi-linear interpolation is applied. Use extrapolation if $k'_j(b,k,s)$ is below (above) k^1 (k^N) and also if $\hat{b}'_j(b,k,s)$ is above b^M, but for $\hat{b'}_j(b,k,s) < b^1$ evaluate the functions at b^1, because the lower bound on bonds represents an ad-hoc debt limit commonly used for calibration of the model to the data (see Durdu et al. [2019]). Note also that, because of the fractional exponent (since typically $\sigma > 1$) the above equation solves only if $c_j(\cdot) - \frac{L_j(\cdot)^{\omega}}{\omega} > 0$, but if this is true for the consumption and labor decision rules implied by the initial conjectures set for the first iteration $(c_0(\cdot), L_0(\cdot))$, it will also be true at any iteration $j > 0$.

3.3 Solve for $b'_{j+1}(b,k,s)$ using the resource constraint:

$$\begin{aligned} b'_{j+1}(b,k,s) = R \{ y_{j+1}(b,k,s) - p v_{j+1}(b,k,s) - \phi(R-1) \left[(1+\tau) L_{j+1}(b,k,s)^{\omega} + p v_{j+1}(b,k,s) \right] \\ - \tilde{i}_j(b,k,s) - (1+\tau) c_{j+1}(b,k,s) + b \} \end{aligned} \tag{31}$$

3.4 Evaluate if the collateral constraint binds. If:

$$\frac{b'_{j+1}(b,k,s)}{R} - \phi R \left[(1+\tau) L_{j+1}(b,k,s)^{\omega} + p v_{j+1}(b,k,s) \right] + \kappa \hat{q}_j(b,k,s) k'_j(b,k,s) \geq 0, \tag{32}$$

the constraint does not bind at the point (b,k,s), the functions with $j+1$ subscripts are saved, and skip to **Step 5**. Otherwise, the constraint binds at this point, the functions with $j+1$ subscripts are discarded and move to **Step 4**.

Step 4. Solve for *new* decision rules when the collateral constraint binds. Since $\hat{q}_j(b,k,s)$ has not changed, we use the same iteration-j implied decision rule for capital $k'_j(b,k,s) = \frac{k}{a}\left[\hat{q}_j(b,k,s) - 1 + a\right]$ and the same function $\tilde{i}_j(b,k,s)$ as before. This is the most computationally intensive step, because it solves a non-linear simultaneous equations system to determine $L_{j+1}(b,k,s), v_{j+1}(b,k,s), c_{j+1}(b,k,s), b'_{j+1}(b,k,s), \tilde{\mu}_{j+1}(b,k,s)$. The five equations in the system are the two optimality conditions for factor allocations, the Euler equation for bonds (with the $\tilde{\mu}$ terms), the credit constraint holding with equality, and the resource constraint. To make the solution more tractable, we express $L_{j+1}(b,k,s), v_{j+1}(b,k,s), c_{j+1}(b,k,s), b'_{j+1}(b,k,s)$ as functions of $\tilde{\mu}(b,k,s)$, and use the results to reduce the system to a single non-linear equation in $\tilde{\mu}(b,k,s)$. In the simplified system, factor allocations, consumption and bonds are functions denoted $L_{j+1}(b,k,s,\tilde{\mu}), v_{j+1}(b,k,s,\tilde{\mu}), c_{j+1}(b,k,s,\tilde{\mu}), b'_{j+1}(b,k,s,\tilde{\mu})$, but to make the notation simpler we write them as depending on $\tilde{\mu}$ only (still, keep in mind the set of equations needs to be solved for each (b,k,s) for which the constraint was found to be binding in step **3.4**):

$$v(\tilde{\mu}) = \left\{ \frac{Ak^{\gamma}\eta^{\frac{\omega-\alpha}{\omega}} \frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}}\left[1 + \phi(R-1) + \tilde{\mu}\phi R\right]} \right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}} \tag{33}$$

$$L(\tilde{\mu}) = \left\{ \frac{\alpha}{\eta(1+\tau)} pv(\tilde{\mu}) \right\}^{\frac{1}{\omega}} \tag{34}$$

$$\frac{b'(\tilde{\mu})}{R} = -\kappa\hat{q}_j k'_j + \phi R pv(\tilde{\mu})\left[1 + \frac{\alpha}{\eta}\right] \tag{35}$$

$$(1+\tau)c(\tilde{\mu}) = Ak^{\gamma}L(\tilde{\mu})^{\alpha}v(\tilde{\mu})^{\eta} - pv(\tilde{\mu}) - \phi(R-1)pv(\tilde{\mu})\left[1 + \frac{\alpha}{\eta}\right] - \tilde{i}_j - \frac{b'(\tilde{\mu})}{R} + b \tag{36}$$

The equations for labor and inputs follow from combining the borrowing constraint with the optimality conditions equating marginal products with marginal costs, including the $\tilde{\mu}$ terms. They are the same equations used in **Step 2**, but now we need to find the value of $\tilde{\mu}_{j+1}$ that solves them, instead of taking as given $\tilde{\mu}_j$.

In addition to equations (33)-(36), the solution for $\tilde{\mu}_{j+1}(b,k,s)$ must also satisfy the Euler equation for bonds, which can be written as:

$$\tilde{\mu}_{j+1}(b,k,s) = 1 - \frac{\beta RE\left[\left(c_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s') - \frac{L_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')^{\omega}}{\omega}\right)^{-\sigma}\right]}{\left(c(\tilde{\mu}_{j+1}(b,k,s)) - \frac{L(\tilde{\mu}_{j+1}(b,k,s))^{\omega}}{\omega}\right)^{-\sigma}} \tag{37}$$

Notice the numerator in the second term in the right-hand-side still applies fixed-point iteration by computing expected marginal utility using j-dated functions only. The values of $c_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')$ and $L_j(\hat{b}'_j(b,k,s), k'_j(b,k,s), s')$ are again determined by bi-linear interpolation.

Algebraic manipulation of equations (33)-(37) reduces to this non-linear equation in $\tilde{\mu}_{j+1}(\cdot)$

$$
\begin{aligned}
&(1-\tilde{\mu}_{j+1}(\cdot))\left\{C_1^{\frac{\omega}{(1-\eta)\omega-\alpha}}\left[\frac{\alpha}{1+\phi(R-1)+\tilde{\mu}_{j+1}(\cdot)\phi R}\right]^{\frac{\eta\omega+\alpha}{(1-\eta)\omega-\alpha}}\right.\\
&\left.-\left[\frac{\alpha C_1}{1+\phi(R-1)+\tilde{\mu}_{j+1}(\cdot)\phi R}\right]^{\frac{\omega}{(1-\eta)\omega-\alpha}} C_2-\left(\frac{\tilde{i}_j(\cdot)-\kappa\hat{q}_j(\cdot)k'_j(\cdot)-b}{1+\tau}\right)\right\}^{-\sigma} \qquad (38)\\
&=\beta R E\left[\left(c_j(\hat{b}'_j(\cdot),k'_j(\cdot),s')-\frac{L_j(\hat{b}'_j(\cdot),k'_j(\cdot),s')^{\omega}}{\omega}\right)^{-\sigma}\right]
\end{aligned}
$$

where:

$$C_1 \equiv \left(\frac{1}{1+\tau}\right)^{1-\eta} Ak^{\gamma}\left(\frac{\eta}{\alpha p}\right)^{\eta} \qquad (39)$$

$$C_2 \equiv \frac{1}{\omega}+\frac{\eta}{\alpha}+\phi\left(1+\frac{\eta}{\alpha}\right)(2R-1) \qquad (40)$$

Note again that, because of the fractional exponent in the right-hand-side of (38), the equation solves only if $c_j(\cdot)-\frac{L_j(\cdot)^{\omega}}{\omega}>0$. Since the first iteration starts with $\tilde{\mu}_0(\cdot)=0$, any state that yields a binding credit constraint in the first iteration will solve for $\tilde{\mu}_1(\cdot)$ as long as the same condition required for the unconstrained consumption function (eq. (30)) to solve in the first iteration holds, namely that $c_0(\cdot)-\frac{L_0(\cdot)^{\omega}}{\omega}>0$ for the decision rules implied by the initial conjectures set for the first iteration. Moreover, since when the constraint binds it must be true that $0<\tilde{\mu}<1$, it follows from eq. (37) that $c_j(\cdot)-\frac{L_j(\cdot)^{\omega}}{\omega}>0$ will hold for any iteration $j>0$. Once $\tilde{\mu}_{j+1}(b,k,s)$ is solved, the functions $v_{j+1}(b,k,s), L_{j+1}(b,k,s), b'_{j+1}(b,k,s), c_{j+1}(b,k,s)$ are determined using equations (33)-(36), but replacing $\tilde{\mu}$ with $\tilde{\mu}_{j+1}(b,k,s)$. The functions with $j+1$ subscripts are saved, and we move to **Step 5**.

It is important to note that, depending on the structure of the occasionally binding constraint, if $\tilde{\mu}$ can be solved for separately after solving for the allocations, **Step 4** is much easier because *FiPIt* does not require a non-linear solver *anywhere*. For example, if working capital is not in the credit constraint, we can set $b'_{j+1}(b,k,s)/R=-\kappa\hat{q}_j(b,k,s)k'_j(b,k,s)$, and this can be used to determine $c_{j+1}(b,k,s)$ directly from the resource constraint. The implied value of $\tilde{\mu}_{j+1}(b,k,s)$ can then be solved for from the bonds Euler equation. The same applies for a credit constraint set to a constant value, as in Ludwig and Schön [2018], where they used $b_{t+1}\geq 0$. Hence, *FiPIt* can solve models with a large class of occasionally binding constraints without using a non-linear solver at any point, whereas the Ludwig-Schön algorithm needs both the Delaunay interpolation and a non-linear solver when the constraint binds.

Step 5. Return to **Step 3** and repeat $\forall(b,k,s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$. This is necessary before proceeding to compute a new asset pricing function, because the complete set of j+1-dated functions is required.

Step 6. Compute the *new* pricing function $q_{j+1}(b,k,s)$. We describe two ways of doing this:

6.1 The *FiPIt* algorithm proceeds in a manner analogous to fixed-point iteration on the Euler equation for bonds, by applying the new decision rules for $c_{j+1}(\cdot), L_{j+1}(\cdot), b'_{j+1}(\cdot), \tilde{\mu}_{j+1}(\cdot)$ to the Euler equation for capital and solving it so as to obtain the following analytic solution for $q_{j+1}(b,k,s)$:

$$q_{j+1}(b,k,s) = \frac{\beta E_t \left[\left(c_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right) - \frac{L_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)^{\omega}}{\omega} \right)^{-\sigma} \left[d'(\cdot) + \hat{q}_j\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right) \right] \right]}{\left(c_{j+1}(\cdot) - \frac{L_{j+1}(\cdot)^{\omega}}{\omega} \right)^{-\sigma} \left(1 - \kappa \tilde{\mu}_{j+1}(\cdot)\right)} \tag{41}$$

where

$$d'\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right) = \gamma A' k'_j(\cdot)^{\gamma-1} L_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)^{\alpha} v_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)^{\eta} - \delta + \frac{a}{2} \frac{\left(k'_j(b'_{j+1}(\cdot), k'_j(\cdot), s') - k'_j(\cdot)\right)^2}{k'_j(\cdot)^2}$$

The asset price used in the right-hand-side of (41) is the conjecture set in **Step 1**. Since all the functions in the right-hand-side are known, the equation solves directly for $q_{j+1}(b,k,s)$. The values of $c_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right))$, $L_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)$ and $\hat{q}_j\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)$ are determined by bi-linear interpolation. The value of the dividends function $d'(\cdot)$ is obtained by applying bi-linear interpolation to evaluate $L_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)$ and $v_{j+1}\left(b'_{j+1}(\cdot), k'_j(\cdot), s'\right)$ in the marginal product of capital and $k'_j(b'_{j+1}(\cdot), k'_j(\cdot), s')$ in the adjustment cost term. Notice that the decision rule for bonds that sets the value of b_{t+1} at which all these functions are interpolated is a j+1-indexed function, not the j-indexed function used in Steps **3** and **4**, but over the capital dimension we are still using the j-indexed decision rule.

6.2 A variant of the algorithm labeled *Fixed-Point Iteration with Forward Solution (FPIFS)* solves for the new price conjecture by iterating to convergence on the capital Euler equation (i.e. it uses the forward solution of the asset price). Index the iterations on this equation with superscript z, the iterations solve this functional equation problem, always using the $j+1$-dated functions and the multiplier $\tilde{\mu}_{j+1}(\cdot)$ obtained in Steps **3** to **5**:

$$q^{z+1}(b,k,s) = \frac{\beta E_t\left[\left(c_{j+1}\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right) - \frac{L_{j+1}\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right)^{\omega}}{\omega}\right)^{-\sigma}\left[d'(\cdot) + q^z\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right)\right]\right]}{\left(c_{j+1}(\cdot) - \frac{L_{j+1}(\cdot)^{\omega}}{\omega}\right)^{-\sigma}\left(1-\kappa\tilde{\mu}_{j+1}(\cdot)\right)} \tag{42}$$

Iterate until $||q^{z+1}(b,k,s) - q^z(b,k,s)|| \leq \varepsilon^q$ for small ε^q, and if the result converges, the final result sets $q_{j+1}(b,k,s)$. The values of $c_{j+1}\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right))$, $L_{j+1}\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right)$, $d'(\cdot)$ and $q^z\left(b'_{j+1}(\cdot),k'_j(\cdot),s'\right)$ are determined using bi-linear interpolation as in step **6.1**.

Step 7. Check the convergence of the conjectured functions. Convergence requires that for small ε^f the following conditions are satisfied $\forall(b,k,s) \in \mathbf{B}\otimes\mathbf{K}\otimes\mathbf{S}$:

$$|q_{j+1}(b,k,s) - \hat{q}_j(b,k,s)| \leq \varepsilon^f \tag{43}$$

$$|b'_{j+1}(b,k,s) - \hat{b}'_j(b,k,s)| \leq \varepsilon^f \tag{44}$$

$$|\tilde{\mu}_{j+1}(b,k,s) - \hat{\tilde{\mu}}_j(b,k,s)| \leq \varepsilon^f \tag{45}$$

If these conditions hold, the recursive competitive equilibrium has been solved. The level of the multiplier on the credit constraint can then be solved for as follows:

$$\mu_{j+1}(b,k,s) = \tilde{\mu}_{j+1}(b,k,s)\left(c_{j+1}(b,k,s) - \frac{L_{j+1}(b,k,s)^{\omega}}{\omega}\right)^{-\sigma} \tag{46}$$

The accuracy of the solution can then be evaluated by verifying that the equilibrium conditions hold, including computations of the maximum and average absolute values of the errors in the Euler equations of k and b.

If any of the three convergence conditions fails, update the conjectured functions using a convex combination of the last conjectures and the new functions to dampen possible overshooting or speed up convergence. This is conventional practice because there is no guarantee that fixed-point iteration algorithms converge, but when they diverge it is generally because they overshoot the true solution.

Hence, the new conjectures are set as:

$$\hat{x}_{j+1}(b,k,s) = (1-\rho^x)\hat{x}_j(b,k,s) + \rho^x x_{j+1}(b,k,s) \tag{47}$$

for $x = [q, b', \tilde{\mu}]$ and some $0 \leq \rho^x$. Notice that $\hat{x}_j(b,k,s)$ in the right-hand-side of this expression represents the initial conjectures that were used in the current iteration, while $\hat{x}_{+1}j(b,k,s)$ in the left-hand-side denotes the new conjectures for the next iteration. Use $0 < \rho^x < 1$ ($\rho^x > 1$) for the particular function $x(\cdot)$ that is not converging (converging too slowly). Return to **Step 2**, setting $\hat{x}_j(b,k,s)$=$\hat{x}_{j+1}(b,k,s)$, and repeat until convergence is attained.

4 Application to Sudden Stops Model

This Section examines solutions of the Sudden Stops model obtained with a set of Matlab programs we developed to implement the *FiPIt* algorithm. The Matlab codes and an Appendix that explains how the codes execute each of the algorithm steps are available online. All the computations were made using Matlab R2017a on a Windows 10 laptop with an Intel Core i7-6700HQ 2.60GHz four-core chip and 16GB of RAM.

The model's parameter values are taken from Mendoza [2010] and listed in Table 1, including the same baseline value for the collateral coefficient ($\kappa = 0.2$). We also use the same Markov process for the model's three shocks. The only difference, as mentioned earlier, is that instead of using preferences with an endogenous rate of time preference we use standard time-separable expected utility with constant discounting, setting the subjective discount factor at $\beta = 0.92$.

The state space consists of evenly-spaced grids with 72 nodes for bonds and 30 nodes for capital. **K** spans the [654.5,885.5] interval and **B** spans the [-188.6,800.0] interval. Solving with larger grids increases sharply execution time and produces negligibly different results, while solving with smaller grids is faster but yields inaccurate results. The Markov process of the shocks has two realizations for each shock and their values together with the associated 8x8 transition probability matrix approximate the variability, autocorrelation, and contemporaneous correlation of TFP, interest rates and the price of imported inputs in the data (see Mendoza [2010] for details).

To assess the performance of the *FiPIt* algorithm, we computed solutions using *FiPIt* and the *FPIFS* variant, as well as solutions from three other algorithms: *TIFS* replaces the fixed-point iteration solution of the bonds decision rule with a standard time iteration solution that uses a non-linear solver, and solves for the price of capital using the forward solution of the capital Euler equation; *TIFPI* uses again standard time iteration for the bonds decision rule, but solves for the price of capital using the fixed-point iteration approach; and *FTI* is the full time iteration solution in which the Euler equations for capital and bonds are solved as a non-linear equation system. In all these solutions except *FTI*, we found faster convergence by setting the dampening parameters for updating the conjectured functions to 0.3 for the price of capital (0.25 for a scenario with 60 capital nodes) and 1 for bonds and $\tilde{\mu}$. For *FTI* solutions, we kept $\rho^x = 1$ for all three functions, and confirmed that these produces convergence in the smallest number of iterations.

Table 1: Calibrated Parameter Values

Parameter		Value
σ	risk aversion coefficient	2.0
ω	labor elasticity coefficient	1.8461
β	discount factor	0.92
a	capital adjustment cost	2.75
ϕ	working capital parameter	0.2579
δ	depreciation rate	0.088
α	labor share	0.59
η	imported inputs share	0.10
γ	capital share of income	0.31
τ	tax on consumption	0.17
A	average TFP	6.982
κ	collateral coefficient	0.20

4.1 Comparison of Results & Performance Metrics

Table 2 reports long-run moments of the main macro aggregates and performance statistics of the algorithm for the following seven solution scenarios: Columns (1) and (2) are *FiPIt* solutions with capital grids of 60 and 30 nodes respectively, (3) is the *TFIS* solution, (4) is the *FPIFS* solution, (5) is the *TIFPI* solution, (6) is the *FTI* solution and Column (7) shows the results from Mendoza [2010] for reference.

The comparison of Cols. (1) and (2) shows that solving using *FiPIt* with the smaller capital grid has nearly no effect on the results but reduces execution time by a factor of 2.5. The moments reported in Columns (2) to (6) are very similar, and in fact identical up to one or two decimals. Hence, all of the four solution algorithms we tried yield effectively the same results. The results from Mendoza [2010] in Column (7) are qualitatively similar in terms of ranking of volatilities and signs and ranking of correlations and autocorrelations, but quantitatively show more differences. These are due to the different discount factors (Mendoza used endogenous discounting) and the different solution methods (Mendoza solved forcing decision rules to be on the nodes of the grids of bonds and capital, instead of using interpolation, and used value function iteration on a quasi planner's problem with w and q restricted to satisfy the labor supply and investment optimality conditions). The one item that differs sharply is the probability of Sudden Stops, which is about 2.0 percent in Cols. (1)-(6) v. 3.3 percent in Mendoza's paper.[8] This is due to the approximately-continuous decision rules obtained using interpolation in our solutions v. decision rules forced to be on grid nodes in Mendoza's solution. This makes our estimates of the frequency with which $\mu(b,k,s) > 0$, and of the trade balance adjustment implied by the associated $b'(b,k,s)$ in those states, more accurate. In all of our results, the long-run probability of states with $\mu > 0$ is about 2.6 percent, but 23 percent of these states do not yield a sufficiently large increase in the trade balance to classify as a Sudden Stop.

The performance metrics for Columns (1)-(6) reported in panel (b) of Table 2 show that all the solutions have similar accuracy, with small maximum and average absolute-value errors in the Euler equations for bonds and capital. The *FTI* solution yields larger errors, but still this makes little difference in the statistical moments it produces relative to those produced by the other solutions.

In terms of execution time, the *FiPIt* method in Col. (2) dominates the other solution methods by large margins.[9] The absolute speeds will vary widely with hardware and software configurations, but the relative speeds are likely to vary less and the ranking across methods based on this criterion is unlikely to change. Comparing speeds relative to *FiPIt*, which took 810 seconds to run, the second fastest method is *FPIFS* in Col. (4), which took 20 percent longer solve. This algorithm only differs from *FiPIt* in that it solves for the price of capital by solving forward the capital Euler equation. The slowest methods are the three that use time iteration (i.e. a non-linear solver) for at least one Euler equation. In Cols. (3) and (5) the bonds decision rule is solved with the time iteration method, but the price of capital is solved using the forward solution in Col. (3) v. fixed-point iteration in Col. (5). This makes little difference in execution time, as they take 4.6 and 5.1 times longer than the *FiPIt* solution in Col. (2), respectively. Interestingly, the standard *FTI* method in Col. (6), which solves the simultaneous non-linear Euler equations for bonds and capital, is significantly *faster* than the methods used in Cols. (3) and (5), indicating that solving only one non-linear Euler equation instead of two does not guarantee a faster algorithm. Still, the *FTI* execution time exceeds that of

[8] We applied the same definition of Sudden Stops: coordinates (b,k,s) in which the collateral constraint binds and the trade balance-GDP ratio is at least 2 percentage points above what the RBC model yields.

[9] *FiPIt* has even lower relative execution times than the other methods when solving the RBC model, because it avoids using the non-linear solver completely (see Section 4.2 for details).

the *FiPIt* solution by a factor of 2.5!

The *FTI* solution is faster than the ones in Cols. (3) and (5) because time iteration takes advantage of the contraction mapping properties of the two non-linear Euler equations by solving them simultaneously while fixed-point iteration methods do not. Intuitively, every iteration with *FTI* tends to generate relatively more accurate outcomes, and hence attains convergence in 94 iterations. The algorithms in Cols. (3) and (5) take more than twice as many iterations (190 iterations for *TIFS* and 207 for *TIFPI*), and still in each they have to use a root finder because they solve for the bonds decision rule using time iteration. The *FiPIt* method converges in a similar number of iterations (196) as these two methods, but goes through each iteration much faster because it avoids using non-linear solvers when the constraint does not bind, overcoming the drawback of not taking advantage of the contraction mapping properties of the Euler equations, and this makes it the fastest method.[10] *FPIFS* in Col. (4) is the second fastest for a similar reason, and it is slower than *FiPIt* because solving the price of capital with the forward solution is slower than with fixed-point iteration.

[10]This suggests that *FiPIt* can be again much faster than *FTI* in applications in which, as explained in Section 3, the structure of the occasionally binding constraint is such that *FiPIt* does not need a root-finder in states in which the constraint binds (e.g. $q_t^b b_{t+1} \geq -\kappa q_t k_{t+1}$, $q_t^b b_{t+1} \geq -\varphi$). We show results for a case like this in subsection 4.2.

Table 2: Long-run Moments & Performance Metrics: Sudden Stops Model ($\kappa = 0.2$)

	(1) FiPIt-large k grid	(2) FiPIt	(3) TIFS	(4) FPIFS	(5) TIFPI	(6) FTI	(7) Mendoza [2010]
			(a) Long-run moments				
Mean							
gdp	393.629	393.619	393.626	393.618	393.626	393.549	388.339
c	273.910	274.123	274.074	274.124	274.073	274.011	267.857
i	67.482	67.481	67.484	67.481	67.484	67.459	65.802
nx/gdp	0.016	0.015	0.015	0.015	0.015	0.015	0.024
k	765.191	765.171	765.202	765.170	765.202	764.922	747.709
b/gdp	0.007	0.015	0.013	0.015	0.013	0.012	-0.104
q	1.000	1.000	1.000	1.000	1.000	1.000	1.000
leverage ratio	-0.106	-0.102	-0.103	-0.102	-0.103	-0.103	-0.159
v	42.618	42.617	42.618	42.617	42.618	42.609	41.949
working capital	76.660	76.658	76.659	76.658	76.659	76.644	75.455
Standard deviation (in percent)							
gdp	3.91	3.94	3.94	3.94	3.94	3.94	3.85
c	3.95	4.03	4.02	4.03	4.02	4.03	3.69
i	13.33	13.33	13.33	13.33	13.33	13.32	13.45
nx/gdp	2.90	2.94	2.94	2.94	2.94	2.94	2.58
k	4.40	4.49	4.49	4.49	4.49	4.50	4.31
b/gdp	18.72	19.62	19.47	19.62	19.47	19.45	8.90
q	3.20	3.20	3.20	3.20	3.20	3.20	3.23
leverage ratio	8.79	9.22	9.15	9.22	9.15	9.14	4.07
v	5.87	5.89	5.89	5.89	5.89	5.89	5.84
working capital	4.33	4.35	4.35	4.35	4.35	4.36	4.26
Correlation with GDP							
gdp	1.000	1.000	1.000	1.000	1.000	1.000	1.000
c	0.849	0.842	0.844	0.842	0.844	0.844	0.931
i	0.646	0.641	0.641	0.641	0.641	0.641	0.641
nx/gdp	-0.122	-0.117	-0.118	-0.117	-0.118	-0.120	-0.184
k	0.757	0.761	0.761	0.761	0.761	0.761	0.744
b/gdp	-0.133	-0.120	-0.119	-0.120	-0.119	-0.117	-0.298
q	0.400	0.387	0.387	0.387	0.387	0.387	0.406
leverage ratio	-0.125	-0.111	-0.111	-0.111	-0.111	-0.108	0.258
v	0.831	0.832	0.832	0.832	0.832	0.832	0.823
working capital	0.994	0.994	0.994	0.994	0.994	0.994	0.987
First-order autocorrelation							
gdp	0.823	0.825	0.824	0.825	0.824	0.825	0.815
c	0.823	0.830	0.829	0.830	0.829	0.829	0.766
i	0.500	0.501	0.500	0.501	0.500	0.500	0.483
nx/gdp	0.589	0.601	0.598	0.601	0.598	0.598	0.447
k	0.964	0.962	0.962	0.962	0.962	0.962	0.963
b/gdp	0.989	0.990	0.990	0.990	0.990	0.990	0.087
q	0.444	0.447	0.446	0.447	0.446	0.446	0.428
leverage ratio	0.991	0.992	0.992	0.992	0.992	0.992	0.040
v	0.776	0.777	0.777	0.777	0.777	0.777	0.764
working capital	0.800	0.801	0.801	0.801	0.801	0.801	0.777
Prob. of Sudden Stops	1.98%	1.99%	2.03%	1.99%	2.04%	2.05%	3.32%
			(b) Performance metrics				
Bonds Euler Equation							
Max Log10 Abs. Euler Error	-3.58	-3.58	-3.58	-3.58	-3.58	-3.52	-
At Grid Points (b, k, s)	(1, 11, 3)	(1, 6, 3)	(1, 6, 3)	(1, 6, 3)	(1, 6, 3)	(2, 1, 7)	-
Mean Log10 Abs. Euler Error	-14.45	-14.45	-12.41	-14.27	-12.39	-12.35	-
Capital Euler Equation							
Max Log10 Abs. Euler Error	-15.38	-15.37	-15.42	-15.37	-15.42	-4.06	-
At Grid Points (b, k, s)	(72, 1, 7)	(72, 1, 7)	(32, 1, 7)	(72, 1, 7)	(32, 1, 7)	(1, 1, 7)	-
Mean Log10 Abs. Euler Error	-16.22	-16.22	-16.07	-16.04	-16.21	-12.45	-
Grid size ($\#b, \#k$)	(72, 60)	(72, 30)	(72, 30)	(72, 30)	(72, 30)	(72, 30)	(80, 60)
Seconds elapsed	1985	810	3735	956	4136	1986	-
Relative to FiPIt	2.5	1.0	4.6	1.2	5.1	2.5	-
Number of iterations	196	196	190	178	207	94	-

Note: Column (1) and Column (2) are for the *FiPIt* algorithm, fixed-point iteration is used for both the bonds decision rule and the price of capital. Column (3) is for the *TIFS* method, which uses the time iteration method for the bonds decision rule and the forward solution of the capital Euler equation for the price of capital. Column (4) is for the *FPIFS* method, which uses fixed-point iteration for the bonds decision rule and the forward solution of the capital Euler equation for the price of capital. Column (5) is for the *TIFPI* method, which uses time iteration for the decision rule for bonds and fixed-point iteration for the price of capital. Column (6) is for the *FTI* method, which solves the bonds decision rule and the price of capital by solving the Euler equations for bonds and capital as two simultaneous non-linear equations. Sudden Stop states are defined as in Mendoza [2010]: states (b, k, s) such that $\mu(b, k, s) > 0$ and the trade balance-GDP ratio is at least 2 percentage points above its value in the RBC model.

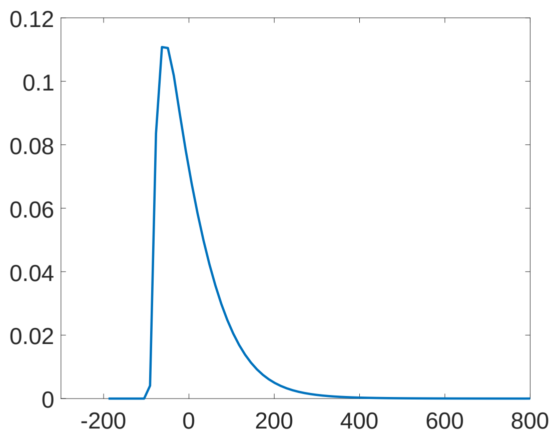
In addition to comparing Euler equation errors, we also compared the recursive equilibrium functions produced by each solution method relative to the *FiPIt* solution. Table 3 shows the maximum and mean of the absolute value of the point-wise differences of the functions as a ratio of the corresponding *FiPIt* solution. The differences are generally negligible, except for the maximum differences for b' and i in the *FTI* solution, which reach 9.94 and 2.25 respectively in states in which the corresponding denominator is very close to zero. Still, as shown in Table 2 this makes little difference in first moments and is nearly irrelevant for second- and higher-order moments.

Table 3: Absolute Values of Differences in Equilibrium Functions Relative to *FiPIt* Solution

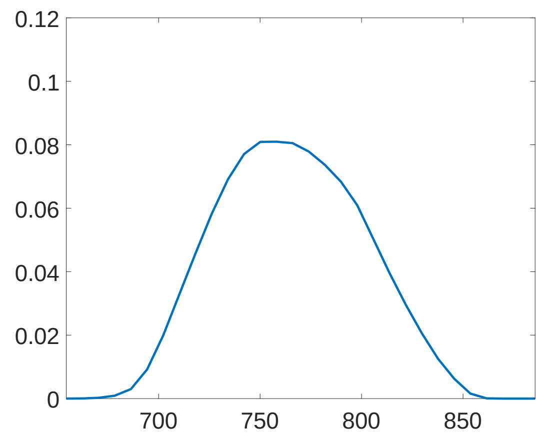
	Differences Relative to *FiPIt* Method			
	(1) TIFS	(2) FPIFS	(3) TIFPI	(4) FTI
Max Difference				
b'	3.17e+00	5.71e-02	3.21e+00	9.94e+00
k'	2.23e-04	7.17e-07	2.23e-04	1.10e-02
q	6.04e-04	1.99e-06	6.04e-04	1.30e-01
c	9.72e-05	7.08e-07	9.72e-05	1.09e-02
i	8.75e-02	5.79e-05	8.74e-02	2.25e+00
L	4.92e-05	3.94e-07	4.92e-05	2.57e-03
v	9.08e-05	7.27e-07	9.08e-05	4.76e-03
gdp	3.27e-05	2.61e-07	3.27e-05	1.77e-03
Mean Difference				
b'	6.29e-04	1.10e-05	6.37e-04	4.35e-03
k'	1.13e-05	2.76e-07	1.15e-05	9.81e-05
q	3.16e-05	7.61e-07	3.22e-05	5.64e-04
c	2.00e-05	3.22e-07	2.03e-05	7.10e-05
i	1.74e-04	3.27e-06	1.76e-04	1.51e-03
L	1.22e-06	3.27e-09	1.22e-06	2.38e-05
v	2.24e-06	6.03e-09	2.25e-06	4.40e-05
gdp	8.16e-07	2.19e-09	8.18e-07	1.62e-05

Figure 1 shows the ergodic marginal distributions of bonds and capital, and the ergodic joint marginal distribution of both variables produced by the *FiPIt* solution. These plots are generated using the full ergodic distribution of (b, k, s), which *FiPit* computes using a procedure that iterates to convergence on the law of motion of the conditional distribution of (b, k, s) (starting from an arbitrary initial condition) taking into account the fact the the decision rules of capital and bonds are generally off the nodes of the corresponding grids. Full details are provided in the Appendix.

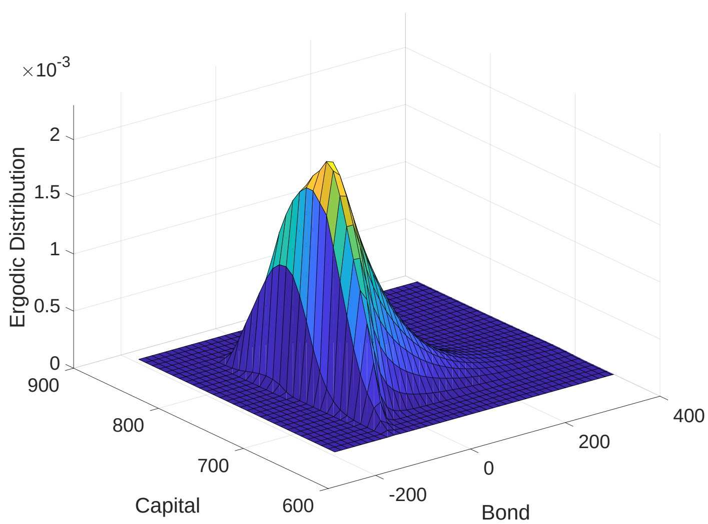
The long-run moments listed in Table 2 were produced using this distribution. The distributions produced by all the other solution methods are visually identical, and hence we only show the ones for the *FiPIt* case. Relative to the distributions that the RBC model would produce, the distribution of bonds shifts to the right because of the credit constraint and the stronger precautionary saving incentives. The distribution of capital shows higher dispersion and a fatter left tail because of the fire-sales of capital in states in which the constraint binds.



(a) Ergodic Bond Distribution



(b) Ergodic Capital Distribution

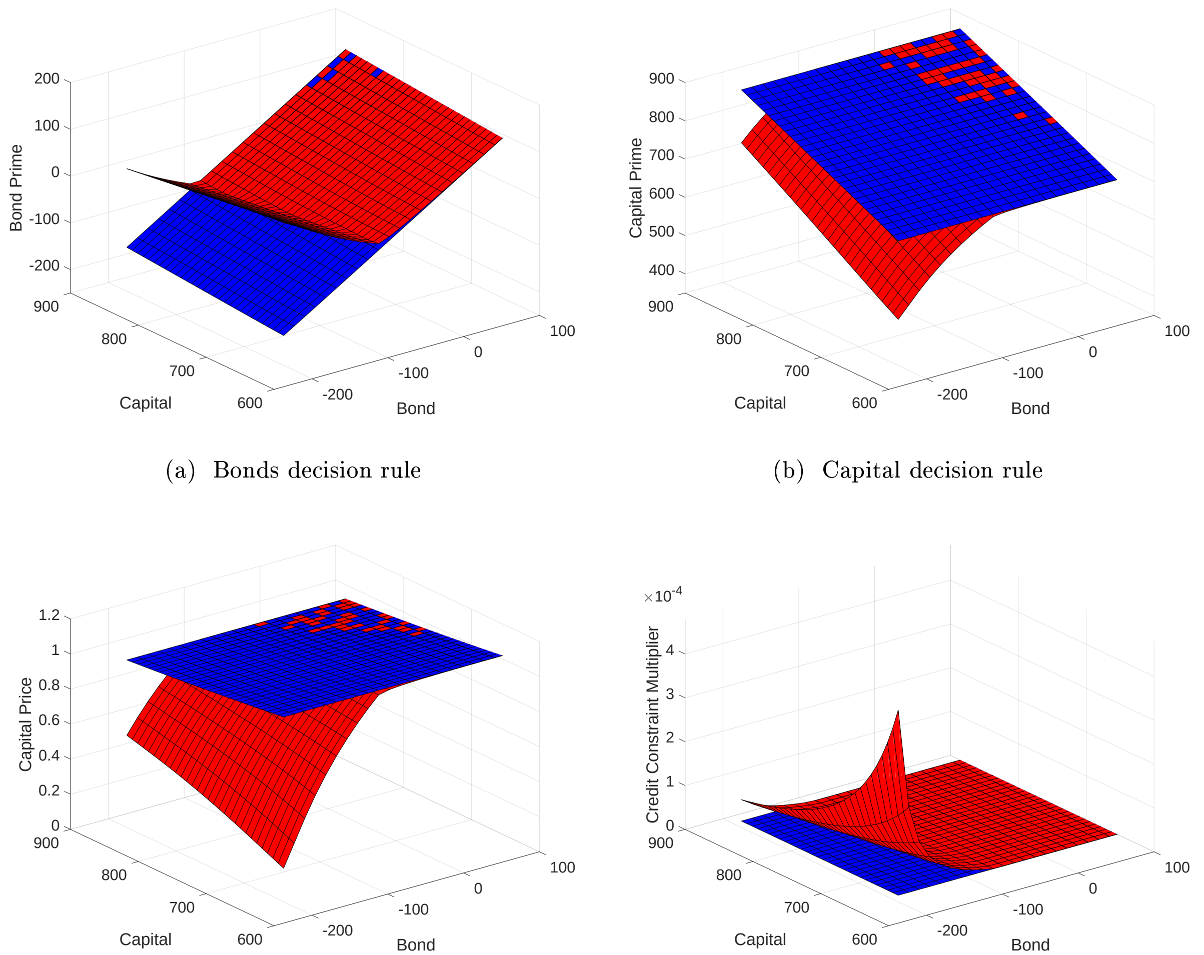


(c) Ergodic Distribution of Bonds and Capital

Figure 1: Long-run Distributions of the Sudden Stops Model Solved with FiPIt

We also examined the recursive equilibrium functions to evaluate the relevance of the global solution to capture non-linearities. Figure 2 shows the decision rules of bonds and capital, the pricing function of capital and the multiplier of the credit constraint across the full state space of endogenous states, $\mathbf{B} \otimes \mathbf{K}$, with s evaluated for a state with low TFP, high interest rate, and high input prices. We show results for the Sudden Stops model and for the RBC variant, and provide only the *FiPIt* results because the other methods yield visually identical graphs. The equilibrium functions of the Sudden Stops model show significant non-linearities, whereas the RBC outcomes are approximately linear. The non-linearities result from the fire-sales of capital when the constraint binds, the resulting collapse in the price of capital, and the associated sharp reversal in the bond position as borrowing capacity collapses.

The sharp curvature of these non-linear solutions highlights the advantages of using a finite-state-space solution method, instead of a colocation method, as well as the importance of solving using first-order conditions and approximately-continuous decision rules. Decision rules that capture accurately the non-linearities implied by occasionally binding constraints are critical for quantifying the positive and normative implications of this class of models, including Sudden Stops models. For their positive implications, the magnitude, dynamics and frequency of financial crises depends critically on the behavior of decision rules near and at the constraint. For the normative implications, quantifying the size of distortions induced by the credit constraint and the properties of optimal policies to tackle them hinges critically on how likely and how severely is the credit constraint expected to bind at t+1 in a state in which it does not bind at t (see Bianchi and Mendoza [2018]).



(a) Bonds decision rule

(b) Capital decision rule

(c) Price of capital

(d) Credit Constaint Multiplier

Figure 2: Equilibrium Recursive Functions of the Sudden Stops & RBC Models

Note: All plots show solutions obtained with the *FiPIt* method. Surface plots in red (blue) are for the SS (RBC) model.

The plots of equilibrium functions do not control for whether particular (b, k, s) triples have positive probability in the stochastic steady state. States with zero-probability are irrelevant in the long run, and if this is the case in the region where equilibrium functions are non-linear, the non-linearities would be of less relevance than what the equilibrium functions suggest. To assess this issue, we follow Mendoza [2010] to calculate impact amplification coefficients and report the results in Table 4. These coefficients measure the excess response of macro variables across the Sudden Stops and RBC solutions for each triple (b, k, s), separating the state space into Sudden Stop (SS) and non-Sudden Stop (NSS) regions.[11] The averages shown in the SS and NSS columns of the Table are computed using the limiting distribution of (b, k, s) of the Sudden Stops model. The results in the SS column measure amplification on impact when a crisis occurs. Differences across the SS and NSS columns illustrate asymmetry, namely the amount by which shocks of identical magnitudes generate different effects when the collateral constraint is present and active v. when is not.

Table 4 compares amplification coefficients produced by the *FiPIt* and *FTI* solutions (the other methods yield nearly identical results). The coefficients differ very marginally and in most instances they are the same up to the second decimal. The Table shows that the Sudden Stops model yields significant amplification and asymmetry. Amplification coefficients on factor allocations and output are relatively smaller, because on impact at date-t when the credit constraint binds it can only affect them via its effect on working capital financing and hence on labor and imported inputs. In turn, this is due to the absence of the wealth effect on labor supply implied by the utility function specification and to the fact that the date-t capital stock is pre-determined.

The *FiPIt* method yields more accurate results than those produced by the solution method used in Mendoza [2010]. The results in Table 4 are qualitatively similar to those reported in Table 4 of Mendoza's paper, but quantitatively there are significant differences. Differences in model structure (i.e. endogenous v. exogenous discounting) play some role, but the bulk of the differences is due to differences in the solution methods. Mendoza solved for decision rules forced to be on grid nodes using value function iteration, while *FiPIt* solves for interpolated decision rules and iterates on the model's optimality conditions. *FiPIt* yields coefficients for "supply side" variables (i.e. GDP, labor, imported inputs and working capital) that are smaller, while those for the rest of the variables (particularly investment and the price of capital) are larger. Moreover, for supply-side variables in the NSS region *FiPIt* yields near-zero amplification while Mendoza reports figures in the -0.29 to -0.11 range. The *FiPIt* results are the correct ones because the amplification coefficients for these variables should indeed differ from zero only due to numerical approximation error. Since k is pre-determined at each date t and there is no wealth effect on labor supply, when $\mu(b, k, s) = 0$ the set of optimality conditions is the same in the RBC and Sudden Stops models and in both cases

[11] A triple (b, k, s) belongs in the SS set if the trade balance-GDP ratio in the Sudden Stops model is 2 percentage points or more above its value in the RBC model, otherwise it belongs in the NSS region. The amplification coefficients for each variable at a given (b, k, s) are calculated as differences relative to their values in the RBC model in the same state and expressed in percent of the unconditional mean of the variable also in the RBC model. For variables defined in ratios, the coefficient is the difference in the Sudden Stops model relative to the RBC model.

all supply-side variables depend only on (k, s). The coefficients around -0.11 to -0.29 that Mendoza obtained result from non-trivial numerical approximation errors due to inaccuracies of the solution algorithm when averaging outcomes for states in which the NSS and SS regions are adjacent and in determining the value of $\mu(b, k, s)$ when assigning (b, k, s) triples to the SS and NSS sets.

Table 4: Amplification and Asymmetry of Sudden Stop events

	(1)		(2)	
	FiPIt		FTI	
	SS	NSS	SS	NSS
gdp	-0.777	-0.001	-0.789	-0.001
c	-3.849	-0.255	-3.882	-0.260
i	-24.965	-1.036	-25.384	-1.089
q	-6.090	-0.253	-6.194	-0.266
nx/gdp	4.033	0.233	4.047	0.238
b'/gdp	4.215	0.251	4.229	0.257
k'/gdp	-1.667	-0.105	-1.680	-0.110
$lev.\ ratio$	1.166	0.081	1.167	0.082
L	-1.178	-0.001	-1.196	-0.002
v	-2.146	-0.003	-2.180	-0.003
$w.\ cap$	-2.160	-0.003	-2.193	-0.003

Note: Sudden Stop (SS) states are defined as states in which the collateral constraint binds and the trade balance-GDP ratio in the Sudden Stop model is more than 2 percentage points above the trade balance-GDP ratio of the RBC model. The coefficients are computed as mean differences relative to the RBC model in percent of the RBC unconditional averages.

4.2 Robustness Analysis & Credit Constraint Variations

The last set of experiments evaluates the robustness and stability of the *FiPIt* algorithm by examining its performance relative to the time iteration method for various parameter changes. This is important in light of the potential instability of fixed-point iteration methods. As documented below, the *FiPIt* method remains stable and continues to outperform the *FTI* method in all the experiments. We also provide results for the RBC variant of the model and for variations of the credit constraint for which *FiPIt* does not require using a non-linear solver in states in which the constraint binds and found even larger gains in execution time in both instances.

Tables 5, 6 and 7 show long-run moments and performance metrics obtained by solving the model using the *FiPIt* and *FTI* methods for these parameter changes: (a) removing working capital ($\phi = 0$); (b) lowering the discount factor ($\beta = 0.91$); (c) reducing the collateral coefficient ($\kappa = 0.15$); (d) increasing the collateral coefficient ($\kappa = 0.25$); (e) setting the collateral coefficient so that the constraint never binds ($\kappa \geq 1.0$), which yields the RBC solution; (f) increasing the labor disutility coefficient ($\omega = 2.5$); and (g) increasing the relative risk aversion coefficient ($\sigma = 3.0$). For each parameter variation, the grids of capital and bonds were re-sized to obtain the fastest solution that does not distort the quantitative results, using identical grids for the *FiPIt* and *FTI* solutions. Still, this resulted in grids of about the same dimensions as before: 71 or 72 nodes in **B** and 30 nodes in **K**, except for case (e) with the RBC model, for which 80 nodes in **B** were needed, and case (f) that needed only 62 nodes in **B**.[12]

The dominance of the *FiPIt* method is robust to all these parameter changes, and in all cases the algorithm is stable and yields solutions nearly identical to the *FTI* results. Comparing across the cases in which the root-finder is needed to solve allocations when the credit constraint binds (i.e. excluding case (a)), *FTI* is 2.0 to 6.0 times slower than *FiPIt* depending on which scenario is considered. Comparing v. the scenario in which *FiPIt* does not need the root-finder when the constraint binds (Col. (2) of Table 5), *FTI* is 13.0 times slower, and for solving the RBC model, which also does not need a root-finder (Col. (6) of Table 6), *FTI* is 18.1 times slower. In both of these instances, *FiPIt* solves in about 2 minutes. Moreover, in most cases *FiPIt* did not require changing the values of the dampening parameters for the updates of the decision rule for bonds ($\rho^b = 1$), the credit constraint multiplier ($\rho^\mu = 1$) and the pricing function ($\rho^q = 0.3$).

It is worth noting that the time iteration solutions required about the same number of iterations (between 87 and 100) and execution time in all the experiments except case (f), which has the smaller **B** grid and used about the same number of iterations but solved faster than the other time iteration solutions. There is more variation in both number of iterations and execution times in the *FiPIt* solutions, but the two tend to move together: The slowest solution was for case (b) which took 1,130 seconds and 244 iterations.

For the case without working capital (case (a)), Column (2) shows the results that *FiPIt* yields when the code is modified to take into account that a root-finder is not needed to solve when the credit constraint binds, as explained in Section 3 (since the constraint is now of the form $b'_{j+1}(b,k,s)/R \geq -\kappa\hat{q}_j(b,k,s)k'_j(b,k,s)$). We also solved an additional experiment with an alterna-

[12]When solving the RBC variant of the model, the bonds grid is extended to accommodate larger debt positions that are part of the equilibrium solution. In this case, **B** consists of 80 nodes spanning the [-300.0,800.0] interval. The upper bound is the same as before, but the lower bound of -300 is significantly smaller (relative to -188.6 used in the solutions reported earlier).

tive credit constraint in the same class that does not require a non-linear solver: $b'_{j+1}(b,k,s)/R \geq \varphi$ with φ set one standard deviation below the average of b' in the limiting distribution of the RBC model. These experiments illustrate the large additional gain in speed that *FiPIt* yields when used to solve models with constraints like these. In Case (a), the *FiPIt* solution is obtained in almost one-third of the time taken by the *FiPIt* algorithm that uses the non-linear solver, which implies that *FiPIt* is faster than the time iteration solution by a factor of 13.0 (v. 5.6 with the *FiPIt* algorithm that uses the non-linear solver). In the case with the constraint given by φ, the *FiPIt* solution is faster than the time iteration method by a factor of 17.9.

Table 5: Sudden Stops Model Variations: Working Capital & Discounting

	(a) Working Capital $\phi = 0$			(b) Discount factor $\beta = 0.91$	
	(1) FiPIt	(2) FiPIt (no root-finder when $\mu > 0$)	(3) FTI	(4) FiPIt	(5) FTI
(a) Long-run moments					
Mean					
gdp	406.361	406.361	406.291	368.772	368.159
c	282.681	282.681	282.564	255.219	254.877
i	69.847	69.847	69.824	59.863	59.678
nx/gdp	0.015	0.015	0.015	0.029	0.029
k	792.035	792.035	791.780	678.870	676.802
b/gdp	-0.202	-0.202	-0.205	-0.161	-0.160
q	1.000	1.000	1.000	1.000	1.000
leverage ratio	-0.095	-0.095	-0.097	-0.195	-0.195
v	45.079	45.079	45.072	39.798	39.729
working capital	0.000	0.000	0.000	71.585	71.460
Standard deviation (in percent)					
gdp	3.71	3.71	3.71	3.93	3.93
c	3.82	3.82	3.82	3.91	3.92
i	13.16	13.16	13.16	12.17	12.16
nx/gdp	2.94	2.94	2.93	2.12	2.11
k	4.44	4.44	4.45	4.53	4.54
b/gdp	20.06	20.06	19.88	2.28	2.26
q	3.16	3.16	3.16	2.88	2.88
leverage ratio	9.47	9.47	9.39	0.69	0.68
v	5.42	5.42	5.43	5.97	5.97
working capital	-	-	-	4.38	4.39
Correlation with GDP					
gdp	1.000	1.000	1.000	1.000	1.000
c	0.820	0.820	0.823	0.969	0.969
i	0.593	0.593	0.594	0.713	0.713
nx/gdp	-0.083	-0.083	-0.085	-0.310	-0.311
k	0.775	0.775	0.776	0.754	0.754
b/gdp	-0.070	-0.070	-0.067	-0.093	-0.096
q	0.334	0.334	0.334	0.442	0.441
leverage ratio	-0.076	-0.076	-0.073	-0.024	-0.030
v	0.795	0.795	0.795	0.833	0.833
working capital	-	-	-	0.988	0.988
First-order autocorrelation					
gdp	0.834	0.834	0.834	0.818	0.818
c	0.860	0.860	0.860	0.759	0.759
i	0.501	0.501	0.500	0.330	0.330
nx/gdp	0.608	0.608	0.606	0.068	0.068
k	0.962	0.962	0.962	0.970	0.970
b/gdp	0.991	0.991	0.991	0.423	0.415
q	0.446	0.446	0.446	0.234	0.235
leverage ratio	0.992	0.992	0.992	0.686	0.679
v	0.788	0.788	0.788	0.756	0.757
working capital	-	-	-	0.755	0.756
P(S.S)	1.43%	1.43%	1.49%	39.98%	40.44%
(b) Performance metrics					
Bonds Euler Equation					
Max Log10 Abs. Euler Error	-4.17	-4.17	-4.13	-3.56	-3.51
At Grid Points (b, k, s)	(1, 1, 3)	(1, 1, 3)	(1, 1, 3)	(3, 1, 3)	(3, 1, 3)
Mean Log10 Abs. Euler Error	-15.53	-15.53	-13.25	-18.96	-12.07
Capital Euler Equation					
Max Log10 Abs. Euler Error	15.43	15.43	-6.36	15.40	-4.18
At Grid Points (b, k, s)	(72, 1, 7)	(72, 1, 7)	(1, 1, 7)	(72, 1, 7)	(2, 1, 7)
Mean Log10 Abs. Euler Error	-16.17	-16.17	-12.77	-22.43	-12.23
Grid size $(\#b, \#k)$	(72, 30)	(72, 30)	(72, 30)	(72, 30)	(72, 30)
Seconds elapsed	297	123	1593	1220	2405
Relative to FiPIt	2.4	1.0	13.0	1.0	2.0
Number of iterations	200	200	95	244	87

Note: Columns (1) are *FiPIt* solutions and Columns (3) are time iteration (*FTI*) solutions. Column (2) shows results for the model with $\phi = 0$ obtained with the *FiPIt* algorithm without using a non-linear solver when $\mu > 0$, since it is not needed.

Table 6: Sudden Stops Model Variations: Collateral Coefficient

	(c) Lower Coll. Coeff. $\kappa = 0.15$		(d) Higher Coll. Coeff. $\kappa = 0.25$		(e) RBC Non-Binding Coll. Coeff. $\kappa \geq 1$	
	(1)	(2)	(3)	(4)	(5)	(6)
	FiPIt	FTI	FiPIt	FTI	FiPIt	FTI
			(a) Long-run moments			
Mean						
gdp	393.503	393.433	393.728	393.659	393.847	393.813
c	276.545	276.436	271.926	271.807	264.021	263.871
i	67.442	67.420	67.517	67.495	67.530	67.518
nx/gdp	0.007	0.008	0.022	0.022	0.045	0.046
k	764.752	764.503	765.564	765.316	765.885	765.759
b/gdp	0.109	0.106	-0.070	-0.073	-0.372	-0.377
q	1.000	1.000	1.000	1.000	1.000	1.000
leverage ratio	-0.057	-0.059	-0.142	-0.144	-0.286	-0.288
v	42.604	42.596	42.630	42.622	42.649	42.646
working capital	76.634	76.620	76.681	76.667	76.716	76.710
Standard deviation (in percent)						
gdp	3.91	3.91	3.96	3.96	3.99	3.99
c	3.87	3.87	4.20	4.20	5.15	5.14
i	13.23	13.22	13.43	13.43	13.51	13.51
nx/gdp	2.90	2.89	3.02	3.01	3.53	3.52
k	4.43	4.44	4.54	4.55	4.65	4.65
b/gdp	19.10	18.92	20.73	20.55	30.28	30.04
q	3.18	3.18	3.23	3.23	3.24	3.23
leverage ratio	8.96	8.88	9.76	9.68	14.42	14.31
v	5.87	5.87	5.91	5.91	5.91	5.91
working capital	4.33	4.33	4.38	4.38	4.40	4.40
Correlation with GDP						
gdp	1.000	1.000	1.000	1.000	1.000	1.000
c	0.842	0.844	0.836	0.839	0.773	0.776
i	0.641	0.642	0.640	0.640	0.640	0.640
nx/gdp	-0.085	-0.088	-0.145	-0.148	-0.227	-0.230
k	0.758	0.758	0.763	0.763	0.767	0.767
b/gdp	-0.169	-0.167	-0.072	-0.069	0.090	0.093
q	0.390	0.390	0.385	0.385	0.381	0.381
leverage ratio	-0.165	-0.164	-0.060	-0.057	0.112	0.115
v	0.831	0.831	0.833	0.833	0.834	0.834
working capital	0.995	0.995	0.994	0.994	0.995	0.995
First-order autocorrelation						
gdp	0.823	0.823	0.826	0.826	0.830	0.830
c	0.823	0.822	0.837	0.837	0.885	0.885
i	0.500	0.499	0.501	0.501	0.516	0.516
nx/gdp	0.602	0.600	0.607	0.604	0.711	0.709
k	0.961	0.961	0.962	0.962	0.964	0.964
b/gdp	0.989	0.988	0.991	0.991	0.997	0.997
q	0.447	0.447	0.446	0.446	0.460	0.460
leverage ratio	0.990	0.990	0.993	0.993	0.997	0.997
v	0.776	0.776	0.777	0.777	0.780	0.780
working capital	0.800	0.800	0.802	0.802	0.807	0.808
P(S.S)	2.90%	2.98%	1.34%	1.45%	-	-
			(b) Performance metrics			
Max Log10 Abs. Euler Error	-3.59	-3.53	-3.57	-3.52	-6.93	-6.93
At Grid Points (b, k, s)	(1, 1, 3)	(1, 1, 3)	(1, 2, 3)	(1, 1, 3)	(1, 1, 3)	(1, 1, 3)
Mean Log10 Abs. Euler Error	-14.11	-12.09	-14.96	-12.71	-15.96	-13.33
Capital Euler Equation						
Max Log10 Abs. Euler Error	15.37	-4.22	15.37	-4.14	15.42	-9.91
At Grid Points (b, k, s)	(71, 1, 7)	(1, 1, 7)	(72, 1, 7)	(1, 1, 7)	(2, 1, 7)	(2, 1, 3)
Mean Log10 Abs. Euler Error	-16.28	-12.38	-16.24	-12.56	-16.36	-12.74
Grid size $(\#b, \#k)$	(71, 30)	(71, 30)	(72, 30)	(72, 30)	(80, 30)	(80, 30)
Seconds elapsed	1066	2668	657	1604	100	1808
Relative to FiPIt	1.0	2.5	1.0	2.4	1.0	18.1
Number of iterations	196	94	196	94	201	94

Note: Columns (1) are for the *FiPIt* algorithm. Columns (2) are for the full time iteration method (*FTI*).

Table 7: Sudden Stops Model Variations: Labor Elasticity & Risk Aversion

	(f) Higer Labor Coeff. $\omega = 2.5$		(g) Higher Risk Aversion $\sigma = 3$	
	(1)	(2)	(3)	(4)
	FiPIt	FTI	FiPIt	FTI
		(a) Long-run moments		
Mean				
gdp	110.477	110.441	393.797	393.722
c	78.995	78.897	277.504	277.357
i	18.937	18.924	67.534	67.509
nx/gdp	-0.007	-0.006	0.005	0.005
k	214.735	214.586	765.801	765.531
b/gdp	0.289	0.278	0.138	0.134
q	1.000	1.000	1.000	1.000
leverage ratio	0.027	0.023	-0.044	-0.045
v	11.960	11.956	42.639	42.630
working capital	21.514	21.507	76.697	76.682
Standard deviation (in percent)				
gdp	3.33	3.33	3.97	3.98
c	6.31	6.24	4.35	4.35
i	13.11	13.11	13.46	13.46
nx/gdp	5.39	5.31	3.25	3.24
k	4.32	4.34	4.59	4.61
b/gdp	56.17	55.03	28.69	28.55
q	3.17	3.17	3.23	3.23
leverage ratio	26.48	25.94	13.55	13.49
v	5.40	5.41	5.91	5.92
working capital	3.76	3.77	4.39	4.40
Correlation with GDP				
gdp	1.000	1.000	1.000	1.000
c	0.451	0.463	0.744	0.747
i	0.614	0.614	0.638	0.639
nx/gdp	-0.077	-0.084	-0.059	-0.062
k	0.755	0.756	0.765	0.765
b/gdp	-0.039	-0.034	-0.058	-0.054
q	0.366	0.365	0.381	0.381
leverage ratio	-0.043	-0.037	-0.048	-0.044
v	0.797	0.797	0.834	0.834
working capital	0.992	0.992	0.995	0.995
First-order autocorrelation				
gdp	0.820	0.821	0.828	0.828
c	0.941	0.939	0.894	0.894
i	0.489	0.489	0.506	0.506
nx/gdp	0.857	0.852	0.733	0.731
k	0.960	0.960	0.963	0.963
b/gdp	0.995	0.995	0.995	0.995
q	0.435	0.435	0.450	0.450
leverage ratio	0.995	0.995	0.996	0.996
v	0.768	0.769	0.779	0.779
working capital	0.793	0.794	0.804	0.805
P(S.S)	3.38%	3.66%	1.29%	1.35%
		(b) Performance metrics		
Bonds Euler Equation				
Max Log10 Abs. Euler Error	-3.43	-3.37	-3.82	-3.75
At Grid Points (b, k, s)	(1, 1, 7)	(1, 1, 7)	(2, 1, 7)	(2, 1, 7)
Mean Log10 Abs. Euler Error	-14.27	-13.16	-15.11	-12.41
Capital Euler Equation				
Max Log10 Abs. Euler Error	14.09	-4.46	16.30	-3.90
At Grid Points (b, k, s)	(62, 1, 7)	(1, 1, 7)	(71, 1, 7)	(1, 1, 7)
Mean Log10 Abs. Euler Error	-14.76	-12.52	-17.13	-12.56
Grid size $(\#b, \#k)$	(62, 30)	(62, 30)	(71, 30)	(71, 30)
Seconds elapsed	283	1700	1108	3533
Relative to FiPIt	1.0	6.0	1.0	3.2
Number of iterations	163	98	246	100

Note: Columns (1) are for the *FiPIt* algorithm. Columns (2) are for the full time iteration method (*FTI*).

5 Conclusions

FiPIt is a simple and fast algorithm designed to solve macroeconomic models with two endogenous state variables and occasionally binding constraints using widely used software. The algorithm applies fixed-point iteration on the Euler equations and by doing so it avoids solving the Euler equations as a non-linear system, as with the standard time iteration method, and does not require interpolation of decision rules over irregular grids, as with the endogenous grids method. Analytic solutions are obtained for recursive equilibrium functions in each iteration of the algorithm, and standard bi-linear interpolation for obtaining these analytic solutions remains applicable.

The *FiPIt* algorithm can handle a large class of occasionally binding constraints, including constraints set to fixed values as well as constraints that depend on endogenous variables. If the constraints are such that equilibrium allocations and prices when the constraints bind must be solved jointly with their associated multipliers, *FiPIt* does need a root-finder in states in which the constraint bind, but for a large class of constraints the two can be solved separately and *FiPIt* does not require a non-linear solver anywhere. In contrast, the endogenous grid method requires a root finder whenever the constraint binds.

We documented the performance gains and accuracy of *FiPIt* by comparing the solutions it produces for a Sudden Stops model of a small open economy vis-a-vis solutions obtained with the time iteration method, and hybrid methods that combine fixed-point and time iteration techniques. In addition, we explored the robustness of our algorithm by documenting solutions for seven parameter variations, including an RBC model in which the constraint never binds. The algorithm was coded in Matlab and executed in a standard Windows laptop. In all cases, *FiPIt* produced results nearly identical to time iteration results with large gains in speed and comparable accuracy as measured by Euler equation errors. Time iteration solutions exceeded the execution time of the *FiPIt* solutions by factors of 2.0 to 18.1. The largest gains were obtained in cases in which *FiPIt* does not use root-finders anywhere, which include the RBC solution and a variation of the Sudden Stops model without working capital. In these cases, solving for allocations when the constraint binds does not require a non-linear solver. Time iteration took 18.1 and 13 times longer than *FiPIt* to solve the RBC model and the Sudden Stops model without working capital, respectively. For the baseline Sudden Stops model, which does need the solver to determine allocations when the constraint binds, time iteration took 2.5 times longer than *FiPIt*.

The *FiPIt* algorithm can be extended to other models with two endogenous states, since applying it requires mainly a fixed-point strategy to iterate on recursive functions using Euler equations. In this paper, *FiPIt* was applied to the Euler equation for bonds to solve for the bonds decision rule and to the Euler equation for capital to solve for the price of capital. The Tobin's Q investment optimality condition was then used to determine the decision rule for capital. It is possible to re-arrange the solution in other ways that *FiPIt* may still accommodate, for example conjecturing the bonds and capital decision rules and using the two Euler equations to solve for their updates. Applying these principles to other models with two endogenous state variables so that they can be solved using *FiPIt* seems relatively straightforward. We provide a brief sketch of four examples in the online Appendix.

Performance gains using *FiPIt* are likely to be even larger if the algorithm is coded in languages that are more efficient than Matlab at handling high-dimensional, sequential loops and parallel optimization, such as Julia, Fortran or Python. The large gains in speed and simplicity of the algorithm also open up the possibility of exploring research topics such as Bayesian estimation of models of financial crisis driven by occasionally binding collateral constraints.

References

S Borağan Aruoba, Jesus Fernandez-Villaverde, and Juan F Rubio-Ramirez. Comparing solution methods for dynamic equilibrium economies. *Journal of Economic dynamics and Control*, 30(12): 2477–2508, 2006.

S Borağan Aruoba, Pablo Cuba-Borda, Kenji Higa-Flores, Frank Schorfheide, and Sergio Villalvazo. Piecewise linear approximations and filtering for DSGE models with occasionally-binding constraints. *Mimeo*, 2019.

Javier Bianchi and Enrique G. Mendoza. Optimal time-consistent macroprudential policy. *Journal of Political Economy*, 126(2):588–634, 2018.

Javier Bianchi, Chenxin Liu, and Enrique G Mendoza. Fundamentals news, global liquidity and macroprudential policy. *Journal of International Economics*, 99:S2–S15, 2016.

Emine Boz and Enrique G Mendoza. Financial innovation, the discovery of risk, and the us credit crisis. *Journal of Monetary Economics*, 62:1–22, 2014.

Johannes Brumm and Michael Grill. Computing equilibria in dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control*, 38:142–160, 2014.

Christopher D Carroll. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics letters*, 91(3):312–320, 2006.

Christopher D Carroll. Lecture notes on solution methods for representative agent dynamic stochastic optimization problems. *Mimeo*, 2011.

Wilbur John Coleman. Solving the stochastic growth model by policy-function iteration. *Journal of Business & Economic Statistics*, 8(1):27–29, 1990.

Bora Durdu, Oliver de Groot, and Enrique Mendoza. Global v. local methods in the quantitative analysis of open-economy models with incomplete markets. *Mimeo*, 2019.

Jeremy Greenwood, Zvi Hercowitz, and Gregory W Huffman. Investment, capacity utilization, and the real business cycle. *The American Economic Review*, pages 402–417, 1988.

Luca Guerrieri and Matteo Iacoviello. Occbin: A toolkit for solving dynamic models with occasionally binding constraints easily. *Journal of Monetary Economics*, 70:22–38, 2015.

Thomas Holden. Computation of solutions to dynamic models with occasionally binding constraints. *EconStor Preprints*, (144569), 2016.

Mark Huggett. The risk-free rate in heterogeneous-agent incomplete-insurance economies. *Journal of economic Dynamics and Control*, 17(5-6):953–969, 1993.

Kenneth L Judd. *Numerical methods in economics*. MIT press, 1998.

Lars Ljungqvist and Thomas J Sargent. *Recursive macroeconomic theory*. MIT press, 2012.

Alexander Ludwig and Matthias Schön. Endogenous grids in higher dimensions: Delaunay interpolation and hybrid methods. *Computational Economics*, 51(3):463–492, 2018.

Enrique G Mendoza. The terms of trade, the real exchange rate, and economic fluctuations. *International Economic Review*, pages 101–137, 1995.

Enrique G Mendoza. Sudden stops, financial crises, and leverage. *American Economic Review*, 100 (5):1941–66, 2010.

Enrique G Mendoza and Katherine A Smith. Quantitative implications of a debt-deflation theory of sudden stops and asset prices. *Journal of International Economics*, 70(1):82–114, 2006.

Fabrizio Perri and Vincenzo Quadrini. International recessions. *American Economic Review*, 108 (4-5):935–84, 2018.

Pontus Rendahl. Inequality constraints and euler equation-based solution methods. *The Economic Journal*, 125(585):1110–1135, 2015.