# Exact Present Solution with Consistent Future Approximation: A Gridless Algorithm to Solve Stochastic Dynamic Models

Wouter J. Den Haan, Michal L. Kobielarz and Pontus Rendahl*

December 14, 2015

**Abstract**

This paper proposes an algorithm that finds model solutions at a particular point in the state space by solving a simple system of equations. The key step is to characterize future behavior with a Taylor series expansion of the current period's behavior around the contemporaneous values for the state variables. Since current decisions are solved from the original model equations, the solution incorporates nonlinearities and uncertainty. The algorithm is used to solve the model considered in Coeurdacier, Rey, and Winant (2011), which is a challenging model because it has no steady state and uncertainty is necessary to keep the model well behaved. We show that our algorithm can generate accurate solutions even when the model series are quite volatile. The solutions generated by the risky-steady-state algorithm proposed in Coeurdacier, Rey, and Winant (2011), in contrast, is shown to be not accurate.

*Key Words*: Risky Steady State, Solution Methods

*JEL Classification*: C63, E10, E23, F41

*Den Haan: Centre for Macroeconomics, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, UK and CEPR, London, UK. E-mail: wjdenhaan@gmail.com. Kobielarz: CentER, Tilburg University. E-mail: michal.kobielarz@gmail.com. Rendahl: Centre for Macroeconomics, University of Cambridge. E-mail: pontus.rendahl@gmail.com.

# 1 Introduction

*Carpe diem, quam minimum credula postero*

*(seize the present, trust tomorrow e'en as little as you may)*

The difficulty in solving stochastic dynamic rational expectations models is that agents are forward looking, which means that one cannot determine this period's behavior unless one knows next period's behavior. It is standard practice to find a recursive solution, which means that the model's solution (as a function of the model's state variables) are the same each period. A model solution is then a set of functions with the following property: If this set of functions is used to describe future model outcomes, then this period's model outcomes are described by the same set of functions. That is, a solution is a fixed point in function space.[1]

By contrast, if future model outcomes are known (as a function of next period's state variables), then the problem is much simpler. If the model has $n$ endogenous variables, then finding the solution for a particular set of state variables would require solving a system of $n$ equations in $n$ unknowns. That is, instead of finding a solution in function space, one only has to find a solution in $n$-dimensional Euclidian space. Building on this logic, Den Haan and De Wind (2012) propose to describe future behavior using a simple perturbation approximation and solve for this period's behavior from the original model equations. Since the original model equations are used, this solution does take into account nonlinearities, uncertainty, and any possible interaction between the two. Unfortunately, describing future behavior with a simple perturbation solution can be so inaccurate that this period's choices are inaccurate as well.

The algorithm that we propose eliminates the ad hoc choice of describing future behavior, but still finds the solution by solving a simple system of equations. The idea is the following. Given future behavior, one can solve for this period's behavior. If one can solve for this period's solution, then one can also solve for the derivatives of this period's solution. The key step of our algorithm is to use these derivatives to construct a Taylor series

[1] The arguments of the functions are the model's state variables.

expansion, which is then used to describe next period's behavior. More formally, we solve for model outcomes in each period using a small system of equations that contains the original model equations as well as some additional equations which ensure that next period's behavior is a Taylor series expansion of this period's behavior, around this period's state variables. Since this period's outcomes are solved using the original model equations, they incorporate any possible nonlinearities, uncertainty, and interactions thereof. We refer to our algorithm as the Exact-Today (ET) algorithm since this period's outcomes are an exact solution of the model equations and we only approximate next period's outcomes.

In contrast to perturbation methods, our Taylor series expansion is only used to characterize next period's behavior in the model equations. Actual behavior is solved from the original model equations and incorporates any possible consequences of nonlinearities and/or uncertainty. The algorithm's advantage relative to projection methods is that it does not require constructing a grid. In more complex models, it may be difficult to construct a grid such that all calculations make sense at all nodes on the grid.[2] Another problem with grid-based methods is the curse of dimensionality, in which the complexity of the problem increases exponentially with the number of state variables. In contrast, the complexity of our algorithm only increases linearly with the number of state variables.

There are already quite a few algorithms to solve stochastic dynamic models. To document the usefulness of our algorithm, we implement it using a challenging model. This is the model considered in Coeurdacier, Rey, and Winant (2011) (CRW). The model is cast in partial equilibrium in which an agent faced with stochastic income and stochastic returns decides how much to save and how much to consume. This type of model is often used in open economy macroeconomics to describe small open economies. The difficulty of this model lies not in its size but in that uncertainty is key in keeping the model well behaved. There exist no steady state and savings diverge absent a sufficient amount of uncertainty. Moreover, savings also diverge if uncertainty is too large.

CRW propose a modified perturbation method to solve this model, which entails taking

[2] As discussion in section 2.3, this matters in practice. For example, particular combinations of state variables may make no sense. The model would never end up at those points, but one may not be aware of this when constructing the grid.

a second-order approximation of the Euler equation and finding a consistent perturbation solution. We document that their solution is actually very inaccurate. By contrast, the solution generated by our algorithm is shown to be very close to an accurate projection algorithm that solves the problem using more than eight million nodes.[3]

Section 2 describes and motivates the algorithm. Section 3 describes the model of Coeurdacier, Rey, and Winant (2011), which we use as a test case for our algorithm. Section 4 documents the accuracy of our algorithm and the algorithm proposed in Coeurdacier, Rey, and Winant (2011).

## 2 The ET algorithm

In this section, we describe the method, which is followed by a short subsection describing what one would actually have to program. The method's merits are discussed in the last subsection.

We focus on a class of models that can be represented by the following system of equations:

$$0 = \mathbb{E}_t \left[ f \left( x_{t-1}, x_t, x_{t+1}, z_t, z_{t+1} \right) \right], \tag{1}$$

where $x_t$ contains the endogenous variables, $z_t$ the exogenous random variables, $\mathbb{E}_t [\cdot]$ denotes the expectation operator, and $f(\cdot)$ is a known, typically nonlinear, function.[4] To simplify notation, we focus on the case when both $x_t$ and $z_t$ are scalars, but the method easily generalizes to the multidimensional case. For the same reason, we assume that $z_t$ is a simple AR(1) process. That is,

$$\begin{aligned} z_{t+1} &= \rho z_t + \varepsilon_{t+1}, & (2) \\ \mathbb{E}_t \left[ \varepsilon_{t+1} \right] &= 0. & (3) \end{aligned}$$

[3] Eight million nodes may seem overly prudent. However, the projection method using "only" 441,000 actually performs worse than our proposed algorithm (!), and a very fine grid is therefore necessary in order to adequately assess the accuracy of the various approaches.

[4] When solving stochastic dynamic models, this system of equations is typically the set of first-order and equilibrium conditions. The variables could be transformations of the underlying economic variables such as the logarithm.

### 2.1 The method

The objective is to find a recursive solution to equation (1) of the following form:

$$x_t = g\left(x_{t-1}, z_t\right). \tag{4}$$

Projection methods posit an approximating function, $p\left(x_{t-1}, z_t; \Phi\right)$, where $\Phi$ is a vector containing the coefficients of the approximating function, and find $\Phi$ such that equation (1) holds exactly, or approximately, on a grid for $(x_{t-1}, z_t)$. Perturbation methods also specify a particular functional form and choose the coefficients $\Phi$ such that the derivatives of $p\left(\cdot; \Phi\right)$ are consistent with the implicit solution of equation (1) at the steady state. By contrast, our algorithm, does not restrict the solution of the *current period's* decisions to be of a particular functional form. Instead, it directly solves for $x_t$ *one point at the time,* by solving a simple system of equations including equation (1). By doing so, it preserves the nonlinearities in $f\left(\cdot\right)$ as well as any interaction between nonlinearities in $f\left(\cdot\right)$ and uncertainty about $\varepsilon_{t+1}$.

The reader may think that it is not possible to directly solve for $x_t$ using equation (1), because equation (1) clearly indicates that to solve for $x_t$ one needs to know how $x_{t+1}$ is determined. The underlying principle of ET that makes this possible is that the relationship between $x_{t+1}$ and *next period's* state variables is imposed to be equal to an approximation of the relationship between $x_t$ and *this period's* state variables. When solving for $x_t$, this approximation for $x_{t+1}$ is only used to describe the behavior at the relevant points in the state space, namely the points where the economy could find itself next period. Moreover, it is important to realize that the approximation we use to describe next period's behavior is not fixed, but depends on the period-$t$ state variables. This process is repeated each period, so that the actual outcome in period $t$ is not given by this approximation, but by the solution to the "ET-system" of equations that includes equation (1).

The order of approximation for ET may vary depending on the desired degree of accuracy. With first-order ET, however, the value of next period's choice, $x_{t+1}$, is approximated by the following linear function:[5]

$$x_{t+1} \approx \widetilde{h}\left(x_t, z_{t+1}; \Phi_t\right) = \phi_{0,t} + \phi_{x,t}\left(x_t - \widetilde{x}_t\right) + \phi_{z,t}\left(z_{t+1} - \widetilde{z}_{t+1}\right), \tag{5}$$

where $\Phi_t$ contains the coefficients of the approximation. That is,[6]

$$\Phi_t = \Phi\left(x_{t-1}, z_t\right) \equiv \left[\widetilde{x}_t, \widetilde{z}_{t+1}, \phi_{0,t}, \phi_{x,t}, \phi_{z,t+1}\right]. \tag{6}$$

The five elements of $\Phi_t$ are functions of the state variables, $x_{t-1}$ and $z_t$, and are, thus, time varying. For the ET algorithm it does not matter whether $x_{t-1}$ and $z_t$ are observations in a simulated time series or an arbitrary point in the state space. Therefore, we focus on the recursive representation of the system of equations, that is,

$$\begin{aligned} 0 &= \mathbb{E}\left[f\left(x_{-1}, x, x_{+1}, z, z_{+1}\right)\right], && \text{(7a)} \\ z_{+1} &= \rho z + \varepsilon_{+1}, && \text{(7b)} \\ \mathbb{E}\left[\varepsilon_{+1}\right] &= 0. && \text{(7c)} \end{aligned}$$

Substituting the approximation for $x_{+1}$ given by equation (5) into equation (7a) gives

$$\begin{aligned} 0 &= \mathbb{E}\left[f\left(x_{-1}, x, \widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right), z, z_{+1}\right)\right] \\ &= \mathbb{E}\left[f\left(x_{-1}, x, \widetilde{h}\left(x, \rho z + \varepsilon_{+1}; \Phi\left(x_{-1}, z\right)\right), z, \rho z + \varepsilon_{+1}\right)\right] \\ &= \mathbb{E}\left[F\left(x_{-1}, x, z, \varepsilon_{+1}; \Phi\left(x_{-1}, z\right)\right)\right]. && \text{(8)} \end{aligned}$$

Since $\varepsilon_{+1}$ is integrated out, equation (8) implicitly defines $x$ as a function of $x_{-1}$ and $z$ for a given $\Phi\left(x_{-1}, z\right)$. We denote this function by $h\left(x_{-1}, z\right)$. Thus,

$$0 = \mathbb{E}\left[F\left(x_{-1}, h\left(x_{-1}, z\right), z, \varepsilon_{+1}; \Phi\left(x_{-1}, z\right)\right)\right]. \tag{9}$$

The idea of ET is to choose $\Phi\left(x_{-1}, z\right)$ such that the function that determines $x_{+1}$, i.e., $\widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right)$, is the Taylor series expansion of the (implicit) function that determines this period's choice, $h\left(x_{-1}, z\right)$, around this period's state variables. The approximation $\widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right)$ will be accurate if next period's state variables are not

[5] Higher-order ET is discussed in appendix D.

[6] It is important to distinguish between $x_t$ and $\widetilde{x}_t$ and between $z_{t+1}$ and $\widetilde{z}_{t+1}$. $x_t$ and $z_{t+1}$ are next period's state variables whereas $\widetilde{x}_t$ and $\widetilde{z}_{t+1}$ are coefficients of $\widetilde{h}\left(\cdot\right)$.

that different from this period's state variables. This is a much weaker condition than the one that will ensure accuracy of the perturbation solution, which is that both this period's and next period's state variables are close enough to their steady state values *and* uncertainty is close to zero.

Since $\widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right)$ is a Taylor series expansion of $h\left(\cdot\right)$ around $x_{-1}$ and $z$, it follows that

$$\begin{aligned} \widetilde{x} &= x_{-1}, && (10a) \\ \widetilde{z}_{+1} &= z. && (10b) \end{aligned}$$

That is, $\widetilde{x}$ and $\widetilde{z}_{+1}$ are equal to $x_{-1}$ and $z$ respectively. Of course, the values of the coefficients $\phi_0$, $\phi_x$, and $\phi_z$ also depend on $x_{-1}$ and $z$. To understand how these coefficients are determined, notice that

$$\begin{aligned} \widetilde{h}\left(\widetilde{x}, \widetilde{z}; \Phi\left(x_{-1}, z\right)\right) &= \phi_0, && (11a) \\ \frac{\partial \widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right)}{\partial x} &= \phi_x, && (11b) \\ \frac{\partial \widetilde{h}\left(x, z_{+1}; \Phi\left(x_{-1}, z\right)\right)}{\partial z_{+1}} &= \phi_z. && (11c) \end{aligned}$$

Differentiating equation (9) with respect to $x_{-1}$ and $z$, gives

$$\begin{aligned} 0 &= \mathbb{E}\left[\begin{array}{c} \frac{\partial F(x_{-1},x,z,\varepsilon_{+1};\Phi(x_{-1},z))}{\partial x_{-1}} \\ +\frac{\partial F(x_{-1},x,z,\varepsilon_{+1};\Phi(x_{-1},z))}{\partial x}\frac{\partial h(x_{-1},z)}{\partial x_{-1}} \end{array}\right], && (12a) \\ 0 &= \mathbb{E}\left[\begin{array}{c} \frac{\partial F(x_{-1},x,z,\varepsilon_{+1};\Phi(x_{-1},z))}{\partial z} \\ +\frac{\partial F(x_{-1},x,z,\varepsilon_{+1};\Phi(x_{-1},z))}{\partial x}\frac{\partial h(x_{-1},z)}{\partial z} \end{array}\right]. && (12b) \end{aligned}$$

From this equation, we obtain expressions for $\partial h\left(x_{-1}, z\right)/\partial x_{-1}$ and $\partial h\left(x_{-1}, z\right)/\partial z$. From now on, we suppress the dependence of $\Phi\left(x_{-1}, z\right)$ on $x_{-1}$ and $z$, although this dependence is a key feature of ET. The key approximation step of ET is to assume that the coefficients of the approximation for next period's choices are set as follows:

$$\begin{aligned}
\phi_0 &= \widetilde{h}\left(\widetilde{x},\widetilde{z};\Phi\right) = h\left(x_{-1},z\right), & \text{(13a)}\\
\phi_x &= \left.\frac{\partial\widetilde{h}\left(x,z_{+1};\Phi\right)}{\partial x}\right| = \left.\frac{\partial h\left(\widehat{x}_{-1},\widehat{z}\right)}{\partial\widehat{x}_{-1}}\right|_{\substack{\widehat{x}_{-1}=x_{-1},\\ \widehat{z}=z}}, & \text{(13b)}\\
\phi_z &= \left.\frac{\partial\widetilde{h}\left(x,z_{+1};\Phi\right)}{\partial z_{+1}}\right| = \left.\frac{\partial h\left(\widehat{x}_{-1},\widehat{z}\right)}{\partial\widehat{z}}\right|_{\substack{\widehat{x}_{-1}=x_{-1},\\ \widehat{z}=z}}. & \text{(13c)}
\end{aligned}$$

In words, the function determining *next period's* choice, $x_{+1}=\widetilde{h}\left(x,z_{+1};\Phi\right)$, is equal to a local approximation of *this period's* choice, $x=h\left(x_{-1},z\right)$, around this period's values of the state variables. Equation (13a) immediately implies that $\phi_0=x$. This leaves us with the following system of three equations to solve for $x$, $\phi_x$, and $\phi_z$:[7]

$$\begin{aligned}
0 &= \mathbb{E}\left[F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)\right], & \text{(14a)}\\
0 &= \mathbb{E}\left[\frac{\partial F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)}{\partial x_{-1}}+\frac{\partial F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)}{\partial x}\phi_x\right], & \text{(14b)}\\
0 &= \mathbb{E}\left[\frac{\partial F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)}{\partial z}+\frac{\partial F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)}{\partial x}\phi_z\right]. & \text{(14c)}
\end{aligned}$$

Lastly, we have to deal with the expectations operator. If $\varepsilon_{+1}$ has discrete support, then the expectations operator can be replaced by a sum of the possible outcomes, $\varepsilon_j$, and the associated weights, $\pi_j$, with $j\in\{1,\cdots,J\}$. If $\varepsilon_{+1}$ has continuous support, then one can use a numerical integration procedure, which boils down to doing the same. For example, if $\varepsilon_{+1}$ has a Normal distribution with mean zero and standard deviation equal to $\sigma$, then using Gauss-Hermite quadrature implies that

$$\begin{aligned}
\varepsilon_j &= \sqrt{2}\sigma\zeta_j, \text{ and} & \text{(15)}\\
p_j &= \frac{\omega_j}{\sqrt{\pi}}, & \text{(16)}
\end{aligned}$$

where the $\zeta_j$'s and the $\omega_j$'s are the Gauss-Hermite nodes and weights, respectively.[8]

[7] Although $\widetilde{x}=x_{-1}$ and $\widetilde{z}_{+1}=z_t$, these variables should be treated as constant coefficients when differentiating $F\left(x_{-1},x,z,\varepsilon_{+1};\Phi\right)$.

[8] Intuitively, one can think of the $\sqrt{2}\sigma\zeta_j$ terms as the possibly outcomes of $\varepsilon_{+1}$ and of the $\omega_j/\sqrt{\pi}$ terms as the associated probabilities. Simple subroutines exist to generate the Gauss-Hermite quadrature nodes and weights. Numerical integration techniques are very powerful. For example, if $\varepsilon_{+1}$ is distributed according

**Relationship to Den Haan and De Wind (2012)** Den Haan and De Wind (2012) propose to set $\widetilde{h}\,(x, z_{+1}; \Phi)$ equal to a standard perturbation solution. Equation (8) can then also be used to solve for $x$ for given values of $x_{-1}$ and $z$. In this approach, the value of $\Phi$ is time invariant and independent of $x_{-1}$ and $z$. The solution for $x$ would still take into account the nonlinearity of $f\,(\cdot)$ and the uncertainty about $\varepsilon_{+1}$. However, the standard perturbation solution may not be a reliable representation of the solution for $x_{+1}$, because it is a local approximation around the steady state values of the state variables when there is no uncertainty. By contrast, ET sets the choice for $x_{+1}$ equal to the behavior according to a local approximation around this period's state variables and uncertainty is not reduced to zero.[9]

## 2.2 What actually needs to be programmed

The key step is to construct a function that defines the three error terms associated with equation (14) as a function of the arguments, $x$, $\phi_x$, and $\phi_z$ for given values of $x_{-1}$ and $z$. Those error terms are given by

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{J} F\left(x_{-1}, x, z, \varepsilon_j; \Phi\right) p_j \\ \sum_{j=1}^{J} \left( \frac{\partial F(x_{-1},x,z,\varepsilon_j;\Phi)}{\partial x_{-1}} + \frac{\partial F(x_{-1},x,z,\varepsilon_j;\Phi)}{\partial x} \phi_x \right) p_j \\ \sum_{j=1}^{J} \left( \frac{\partial F(x_{-1},x,z,\varepsilon_j;\Phi)}{\partial z} + \frac{\partial F(x_{-1},x,z,\varepsilon_j;\Phi)}{\partial x} \phi_z \right) p_j \end{bmatrix}, \tag{17}$$

whereas before

$$F\left(x_{-1}, x, z, \varepsilon_j; \Phi\right) = f\left(x_{-1}, x, \widetilde{h}\left(x, \rho z + \varepsilon_j; \Phi\right), z, \rho z + \varepsilon_j\right), \tag{18}$$

$$\widetilde{h}\left(x, \rho z + \varepsilon_j; \Phi\right) = \phi_0 + \phi_x\left(x - \widetilde{x}\right) + \phi_z\left(z_{+1} - \widetilde{z}\right), \tag{19}$$

and

$$\Phi = [\widetilde{x}, \widetilde{z}, \phi_0, \phi_x, \phi_z] = [x_{-1}, z, x, \phi_x, \phi_z]. \tag{20}$$

to a Normal distribution, and one wants to calculate $\mathbb{E}\left[F\left(\varepsilon_{+1}\right)\right]$, then the Gauss-Hermite quadrature approximation with $J$ nodes gives the exact for *all* $(2J-1)^{\text{th}}$-order polynomials, and an accurate answer when $F\left(\cdot\right)$ is well approximated by a $(2J-1)^{\text{th}}$ order polynomial.

[9] The procedure used in Den Haan and De Wind (2012) improves substantially upon standard perturbation solutions for some models. However, it is shown to be only slightly better than the regular perturbation solution for others.

Defining these error terms requires taking derivatives. As shown in appendices D and F, this is a very simple programming step if one has access to a symbolic toolbox even if one considers higher-order approximations. For quite a few problems this could also be done by hand.

The next and final step is to use an equation solver to find the values of $x$, $\phi_x$, and $\phi_z$ that set the three error terms equal to zero for given values of $x_{-1}$ and $z$.

## 2.3 Merits of the ET algorithm

The ET algorithm shares with projection methods the property that the numerical solution incorporates the uncertainty and nonlinearity underlying the original system of equations without modifying $f(\cdot)$ and with either no modification of the expectations operator (in case $\varepsilon_{+1}$ has discrete support) or with a minor modification (in case $\varepsilon_t$ has continuous support). This is even true for first-order ET, since the period-$t$ choice variables are always solved from the original nonlinear set of first-order conditions. The approximation aspect of the ET solution *only* affects the characterization of *next period's* policy functions. Moreover, the derivatives of the Taylor-series expansion used to characterize next period's behavior vary with the value of this period's state variables. By contrast, the traditional perturbation method characterizes this period's behavior with a particular *fixed* Taylor series expansion using the state variables *and* the amount of uncertainty as its arguments. The derivatives used in this Taylor-series approximation correspond to the derivatives at the steady state. This means that the accuracy of the standard perturbation solution is only guaranteed if the state variables are close to their steady state values and the model's volatility parameters are close to zero. Higher-order perturbation does incorporate the effect of uncertainty, but imposes that the effect of uncertainty on model outcomes is equal to an extrapolation of the effect of uncertainty on behavior at a situation when there is no uncertainty to begin with.

Projection methods, on the other hand, solve a much larger set of simultaneous equations, namely the model equations at *all* grid points.[10] By contrast, ET can solve for the

[10] If the number of nodes exceeds the number of coefficients, then a minimization routine would be used.

choices at a particular point in the state space without having to solve for the solution at other points in the state space. For the application considered in this paper, which is quite challenging, the projections algorithm has to solve a simultaneous system of more than eight million nodes, whereas second-order ET needs to solve a system of ten equations and ten unknowns.[11] ET can do so because if focuses on behavior at the *relevant* points in the state space, namely those where the economy could be in the next period, and it does so by imposing that next period's choices are determined by a Taylor-series expansion of this period's choices. It is true that ET has to solve this system many times, but it is a fixed system of equations, so it only has to be defined once.

It is important to realize that the comparison of standard projection methods and ET is not only an issue of comparing the speed of solving a large system of equations once versus solving a smaller system of equations many times. There are two potential advantages of ET. The first is that ET is not subject to the curse of dimensionality. Whereas the number of elements of a regular grid increases exponentially with the number of state variables, the number of unknowns the ET algorithm has to solve for increases at a slower rate. In particular, it increases at a linear rate if a first-order approximation is used to describe next period's behavior.[12] A second advantage of ET is related to the points considered in calculating the numerical approximation. To apply standard projection methods, one has to construct a grid and the algorithm must be able to find a sensible solution at *all* grid points. But there may be points in the state space where some calculations are not well defined.[13] This is not a problem if the economy never reaches those points. Unfortunately, when constructing the grid one typically would not know which points in the state space one has to exclude for this reason. This is a relevant problem in practice and has motivated some researchers to develop methods that – like ET – use simulated time paths instead of

[11] First-order ET requires solving a system of three equations in three unknowns.

[12] Recall that even the first-order ET solution takes into account the full nonlinearity and uncertainty of the underlying system.

[13] One possibility is that there is simply no solution at these nodes. Another possibility is that there is no solution for the particular functional form chosen for the approximation. But even if neither is the case, then it still may be the case that some points in the state space are particularly problematic *in the process of* finding the numerical solution, which typically involves considering many values for $\Phi$.

pre-specified grids.[14]

## 3 The model of Coeurdacier, Rey, and Winant (CRW)

We use the algorithm to solve the model considered in Coeurdacier, Rey, and Winant (2011). This is a challenging model to solve, since the model has no steady state and is not well defined when there is no or not a sufficient amount of uncertainty.[15] Moreover, as discussed below, the model can also generate non-stationary behavior if there is too much uncertainty and there are some important nonlinearities.

In the CRW model, an infinitely-lived agent decides how much to consume, $c_t$, and how much funds to invest and carry over to the next period, $w_t$. Both the agent's income, $y_t$, and the rate of return, $r_t$, are exogenous stochastic processes. The solution to the agent's problem is characterized by the following set of equations:

$$\begin{aligned} c_t^{-\gamma} &= \beta \mathbb{E}_t \left[ c_{t+1}^{-\gamma} r_{t+1} \right], && (21) \\ c_t + w_t &= y_t + w_{t-1} r_t, && (22) \\ y_t &= (1-\rho_y)\,\overline{y} + \rho_y y_{t-1} + \varepsilon_{y,t}, \quad \varepsilon_{y,t} \sim N\left(0, \sigma_y^2\right), && (23) \\ r_t &= (1-\rho_r)\,\overline{r} + \rho_r r_{t-1} + \varepsilon_{r,t}, \quad \varepsilon_{r,t} \sim N\left(0, \sigma_r^2\right). && (24) \end{aligned}$$

If there would be no uncertainty, the Euler equation reduces to

$$c_{t+1} = \left(\beta \overline{r}\right)^{1/\gamma} c_t, \qquad (25)$$

which implies that consumption would either increase without bound or go to zero unless

[14] Simulation-projection methods are not new. An example is the Parameterized Expectations Algorithm of Den Haan and Marcet (1990). As pointed out in Judd (1992) and documented in Den Haan (1995), those earlier simulation approaches can be quite inefficient in that accuracy of the projection's part of the algorithm, i.e., the regression phase, requires a very large number of observations. Judd, Maliar, and Maliar (2011) develop a simulation-based method with a more efficient projection's element.

[15] Standard projection methods can be used to solve this model, but it is not a trivial problem for these methods either. Since uncertainty is essential, one cannot follow standard practice of first solving a model with little uncertainty and then gradually moving to the case with the desired levels of uncertainty.

$\beta = 1/\bar{r}$. That is, there is no steady-state solution when $\sigma_y = \sigma_r = 0$.[16] In the CRW model, there are no constraints on $w_t$ and short positions of any size are allowed. Thus, uncertainty is key for the ability of the model to generate a stationary solution.[17]

**Natural borrowing constraint.** For the projections algorithm to work smoothly, we found it necessary to impose the natural borrowing constraint. That is, the maximum amount that an agent can borrow is such that their income will always be enough to cover interest payments, even in the worst possible circumstances. In particular, when solving for the projections solution, we impose that

$$w \geq -\frac{y_{\text{low}}}{r_{\text{high}} - 1}, \tag{26}$$

where $y_{\text{low}}$ is the lowest possible value for $y_t$ and $r_{\text{high}}$ the highest possible value for $r_t$.[18] For the parameters considered, this constraint turns out to be never binding, even in the very long simulations considered. As discussed below, this does not imply that it does not affect the behavior of the agent's choices. Neither the CRW algorithm nor the ET algorithm impose the natural borrowing constraint. Nevertheless, as discussed below, the ET solution is characterized by nonlinear behavior that resembles the nonlinearity induced by the natural borrowing constraint.

**Parameter values.** The parameter values are reported in table 1. The two key parameters are the standard deviations of the innovations to output and the rate of return. As documented below, the model variables are quite volatile at the chosen values. Nevertheless, the values for $\sigma_r$ and $\sigma_y$ used here are lower than those chosen in Coeurdacier, Rey, and Winant (2011). We do not use the original CRW values for our comparison, because our accurate projection algorithm indicates that the model solution is not well

[16] When $\beta = 1/\bar{r}$, then there would be a steady state with $c_t = \bar{y} + (r-1)w_0$, but any amount of uncertainty would cause the solution to be non-stationary.

[17] In appendix A, we provide some intuition for this statement. A more formal analysis of the model's properties can be found in Chamberlain and Wilson (2000).

[18] We set $y_{\text{low}}$ equal to $\bar{y}$ minus four standard deviations of the unconditional standard deviation of $y_t$ and we set $r_{\text{high}}$ equal to $\bar{r}$ plus four standard deviations of the unconditional standard deviation of $r_t$.

behaved at those values. As documented in E, it displays "escape dynamics". That is, the simulated time paths for wealth will occasionally – but for sustained time periods – diverge to *extremely* high values.[19] The CRW method imposes stationarity even when it does not make sense to do so. Although less restrictive, the ET algorithm also does not seem capable of capturing such explosive behavior.[20] Therefore, we chose more moderate volatility levels at which the generated series do not display such divergent behavior. The case with the original CRW parameter values is discussed in more detail in appendix E. We follow CRW and choose the value of $\overline{r}$ such that the risky steady state value for wealth is equal to 0.[21]

## 4 Evaluation of solution methods

In this section, we compare the solutions obtained with the CRW and ET algorithms with an accurate projections-method solution.[22] The CRW procedure is identical to the one used in Coeurdacier, Rey, and Winant (2011). The procedure consist of deriving a second-order approximation of the Euler equation and a perturbation solution that are consistent with each other.[23] To implement the ET procedure, we use a second-order approximation to describe next period's choices.[24] The reason we do not use a first-order approximation is that the ET system of equations does not have a solution at some points in the sample

[19] Our projection method imposes an upper bound on wealth. Without imposing such an upper bound, the solution diverges to infinity. Outside the (finite) grid, the solution cannot be relied on to be accurate. Our experience indicates that the simulated wealth series reach this upper bound *even* when it is set at extremely high values for a sufficiently long sample. The stochastic innovations for $r_t$ and $y_t$ have a normal distribution and could in principle take on extremely high values. This is not the cause of the divergence, since the same explosive behavior is observed when the values of $r_t$ and $y_t$ are restricted to be in a bounded set.

[20] With the second-order version, the method fails to find a solution at some point during the simulation. It might still be possible that a higher-order version of the algorithm would be successful.

[21] Although, the original system of equations does not have a steady state, the second-order approximation used by CRW does have a steady state, which is referred to as the risky steady state.

[22] The accuracy of the projections-method solution is documented in appendix B.

[23] A description of the CRW procedure is given in appendix C.

[24] Details of the implementation of the ET algorithm are given in appendix D.

when only a first-order approximation is used.

**Policy functions.** Figure 1 plots the policy functions for the change in wealth, $w - w_{-1}$, as a function of beginning of period wealth, $w_{-1}$, for different values of the realized rate of return and one particular value of income.[25] The top panel displays the results for both the ET and the accurate projection solution. The solutions of these two algorithms are very close for most values of $w_{-1}$. The only noticeable differences occur at values of beginning-of-period wealth close the natural borrowing constraint for higher values of the realized rate of return. The projection solution indicates that $w - w_{-1}$ gets less negative rapidly as $w_{-1}$ is close to the natural borrowing constraint. The ET solution only uses local information and it does not impose the natural borrowing constraint at all. It is, therefore, remarkable that the ET solution displays a similar nonlinearity close to the natural borrowing constraint.[26]

The figure documents another – more important – nonlinearity: The slope of the policy function for $w - w_{-1}$ as a function of $w_{-1}$ depends on the level of this period's realized rate of return, $r$. The slope is positive at high rates of return and negative at low rates of return. The level of $r$ does not only affect this period's resources through $rw_{-1}$, it also affects the expected rate of return. Another way to think of the nonlinearity associated with $r$ is the following: If $r$ increases then $w - w_{-1}$ increases for higher levels of wealth and decreases for lower levels of wealth.[27]

The bottom panel displays the results for the CRW solution. Although the CRW solution is quite accurate for the middle value of $r$ it predicts that $\partial w / \partial w_{-1}$ does not depend on $r$. As discussed above, even the sign of $\partial w / \partial w_{-1}$ depends on the level of $r$ according to the accurate projection solution (and the ET solution).

[25] The results are very similar for other income values.

[26] Although ET does not impose the natural borrowing constraint, it does avoid choosing wealth levels which could imply very low consumption levels next period, which becomes more likely at very low wealth levels.

[27] If $r$ increases, then a saver will have more funds in the current period and a borrower will have less resources. The realization of $r$ also affects the expected return. The associated income effect also differs for savers and lenders.

Figure 2 repeats the exercise for the consumption policy function. The ET policy function accurately captures that the slope of the consumption function depends on the level of $r$. The CRW policy function does not. Moreover, it overpredicts the consumption choice at high rates or return and underpredicts the consumption choice at low rates of return, especially at lower wealth levels. The nonlinearity associated with the natural borrowing constraint is less pronounced in the consumption policy than in the wealth policy function.

**Simulated time paths.** Next we compare simulated time paths. These will reveal possible differences and similarities in exactly those parts of the state space that matter for model properties. The model variables are quite persistent, so we use a long simulation of 80,000 observations. Panels A and B of figure 3 plot the generated values for wealth and consumption, respectively. The figure shows that the ET solutions for consumption and wealth follow the accurate projection solutions closely. That is clearly not the case for the CRW solution. These figures also document that the series are quite volatile even though the chosen standard deviations are less than those used in Coeurdacier, Rey, and Winant (2011).[28]

Figures 4 and 5 display wealth and consumption time paths for four shorter samples taken from the full sample. These figures show in greater detail that the behavior of the CRW solution can be quite different from the accurate projection solution. For example in the top right panel of figure 4, the wealth series of all three algorithms display a similar downward trend in the beginning of the sample. While this downward trend continues according to the projection and ET solutions, the CRW solution displays a remarkable recovery. In the top left panel, the projection and ET solutions display a sharp increase which is not present in the perturbation solution.

**Comovement of CRW & ET time paths with accurate projection solution.** Table 2 reports how close the ET and CRW solutions are to the accurate projection solution using average and maximum absolute deviations as well as correlation coefficients.

[28] To put the volatility in perspective, note that the mean income level is equal to 1.

Whereas the correlation between the time series generated with the ET algorithm and those generated by the accurate projection solution are virtually equal to 1, they are considerably lower for the CRW series. For example, the correlation between the CRW wealth series and the projection wealth series is only 0.729. Despite the almost perfect correlation, the ET data are not exactly equal to their projection equivalent. In particular, there is an average absolute difference of 0.83% for wealth and 0.21% for consumption. These compare very favorable to the outcomes for CRW,which are 52.3% and 13%. The results are similar for the maximum differences from the accurate projection solution. For the ET data, the maximum differences are 3.62% for wealth and 0.92% for consumption, whereas the corresponding numbers are 268.2% and 68.4% for the CRW solution.

**Comparison of generated moments.** Tables 3 and 4 report moments according to the three different solutions for the generated wealth and consumption data, respectively. The statistics based on the projection and the ET algorithm are very close to each other. This is not true for the CRW statistics. In particular, the data generated by the CRW algorithm are substantially less volatile; the standard deviation of wealth is only 53% of the projection analogue and the standard deviation of consumption is only 54% of the projection analogue.

# A The role of uncertainty for stationarity

The following second-order approximation of the Euler equation provides some intuition for the result that the solutions for $c_t$ and $w_t$ could be stationary because of uncertainty.

$$\begin{gathered} c_t^{-\gamma} \approx \beta \mathbb{E}_t \left[ r_{t+1} \right] \mathbb{E}_t \left[ c_{t+1} \right]^{-\gamma} \\ + \\ \gamma \left( \gamma + 1 \right) \beta \mathbb{E}_t \left[ r_{t+1} \right] \mathbb{E}_t \left[ c_{t+1} \right]^{-\gamma - 2} \mathbb{V}_t \left( c_{t+1} \right) \\ - \beta \gamma \mathbb{E}_t \left[ c_{t+1} \right]^{-\gamma - 1} \mathbb{C}_t \left( c_{t+1}, r_{t+1} \right), \end{gathered} \tag{27}$$

where

$$\mathbb{V}_t\left(c_{t+1}\right) \equiv \mathbb{E}_t\left[\left(c_{t+1}-\mathbb{E}_t\left[c_{t+1}\right]\right)^2\right], \tag{28}$$

$$\mathbb{C}_t\left(c_{t+1}, r_{t+1}\right) \equiv \mathbb{E}_t\left[\left(c_{t+1}-\mathbb{E}_t\left[c_{t+1}\right]\right)\left(r_{t+1}-\mathbb{E}_t\left[r_{t+1}\right]\right)\right]. \tag{29}$$

First, consider the case when the rate of return is constant, that is, $\varepsilon_{r,t} = 0 \; \forall t$. This implies that $\mathbb{C}_t\left(c_{t+1}, r_{t+1}\right) = 0$. Also, suppose that the discount rate exceeds the average rate of return, which – by itself – would induce agents to run down their wealth and then take on debt. Uncertainty increases the right-hand side of the Euler equation and dampens the desire of agents to consume more in this period than the next and to reduce wealth levels. Would this mean that the inequality is reversed and that it is optimal to save a lot and increase future consumption levels. The answer is no. The magnitude of the uncertainty effect depends on the value of $\mathbb{E}_t\left[c_{t+1}\right]^{-\gamma-2}$. Consequently, expected consumption is high enough and $\mathbb{E}_t\left[c_{t+1}\right]^{-\gamma-2}$ low enough, then the uncertainty effect is less important.

If the covariance between unexpected changes in $c_{t+1}$ and $r_{t+1}$ is negative—that is, bonds act as a hedge—then the covariance has a similar dampening role. However, if this covariance is positive, then stochastic rates of return would reinforce the desire of the agent to keep on reducing consumption over time.

# B Accuracy of the projections solution

In this section, we document that the solution generated by the projection algorithm is accurate. It is important to establish this, since the projection solution is used as the benchmark against which the other solutions are compared.

The projection solution uses 8,405,000 grid points; 41 for both $r$ and $y$ and 5,000 for $w_{-1}$. We use the Dynamic Euler-equation accuracy test, described in Den Haan (2010). The test compares the generated time series with the series that are obtained by explicitly solving the first-order conditions, using the approximation only to calculate next period's wealth choice.[29] This test is more stringent than the standard Euler-equation accuracy

[29] In particular, the numerical solution is not used to determine the amount of wealth carried over into the next period.

test, since it would detect if small errors accumulate in a simulation. Figure 7 documents that the two series are very close to each other. Table 5 reports some key statistics. Both the figure and the table indicate that the solution is very accurate.

# C The CRW procedure

The CRW procedure starts out with the second-order approximation of the Euler equation, which is given by

$$c_t^{-\gamma} = \beta \left[ \begin{array}{c} \mathbb{E}_t \left[r_{t+1}\right] \mathbb{E}_t \left[c_{t+1}\right]^{-\gamma} \\ +\gamma \left(\gamma+1\right) \beta \mathbb{E}_t \left[r_{t+1}\right] \mathbb{E}_t \left[c_{t+1}\right]^{-\gamma-2} \mathbb{V}_t \left(c_{t+1}\right) \\ -\beta\gamma \mathbb{E}_t \left[c_{t+1}\right]^{-\gamma-1} \mathbb{C}_t \left(c_{t+1}, r_{t+1}\right) \end{array} \right], \tag{30}$$

where $\mathbb{V}_t \left(c_{t+1}\right)$ and $\mathbb{C}_t \left(c_{t+1}, r_{t+1}\right)$ are the variance of $c_{t+1}$ and the covariance of $c_{t+1}$ and $r_{t+1}$, respectively, both conditional on period-$t$ information. In addition, the budget constraint, and the laws of motion for $y_t$ and $r_t$ are needed. That is

$$c_t + w_t = y_t + w_{t-1} r_t, \tag{31}$$

$$y_t = (1-\rho_y)\overline{y} + \rho_y y_{t-1} + \varepsilon_{y,t}, \quad \varepsilon_{y,t} \sim N\left(0, \sigma_y^2\right), \tag{32a}$$

$$r_t = (1-\rho_r)\overline{r} + \rho_r r_{t-1} + \varepsilon_{r,t}, \quad \varepsilon_{r,t} \sim N\left(0, \sigma_r^2\right). \tag{32b}$$

The objective of the CRW procedure is to find an approximation to the savings decision that is linear in levels. The latter is given by

$$w_t = \overline{w} + W_w \widehat{w}_{t-1} + W_y \widehat{y}_t + W_r \widehat{r}_t. \tag{33}$$

Using the budget constraint and this linear approximation for $w_t$, one can derive the second-order terms in the approximated Euler equation. The results are as follows:

$$\mathbb{V}_t \left(c_{t+1}\right) = \left( \left(1-W_y\right)^2 \sigma_y^2 \right) + \left(W_r^2 + w_t^2 - 2 w_t W_r\right) \sigma_r^2, \text{ and} \tag{34}$$

$$\mathbb{C}_t \left(c_{t+1}, r_{t+1}\right) = \left(w_t - W_r\right) \sigma_r^2. \tag{35}$$

The idea of the CRW algorithm is to find a first-order perturbation solution to the system of equations consisting of equations (30), (31), (32), (34), and (35), which is consistent with the definitions of $\mathbb{V}_t\left(c_{t+1}\right)$ and $\mathbb{C}_t(c_{t+1}, r_{t+1})$ given in equations (34) and (35). One could find the values for $\overline{w}$, $W_r$, $W_y$ using an equation solver or with the following iterative procedure. Start with initial values for $W_r$ and $W_y$ and denote these by $\widetilde{W}_r$ and $\widetilde{W}_y$. If we use these values to calculate $\mathbb{V}_t\left(c_{t+1}\right)$ and $\mathbb{C}_t(c_{t+1}, r_{t+1})$ and substitute the expressions for $\mathbb{V}_t\left(c_{t+1}\right)$ and $\mathbb{C}_t(c_{t+1}, r_{t+1})$ into equation (30), we get the CRW system of equations:

$$
\begin{gathered}
c_t^{-\gamma} \\
= \qquad (36) \\
\left[\begin{array}{c}
\beta\mathbb{E}_t\left[r_{t+1}\right]\mathbb{E}_t\left[c_{t+1}\right]^{-\gamma} \\
+\gamma\left(\gamma+1\right)\beta\mathbb{E}_t\left[r_{t+1}\right]\mathbb{E}_t\left[c_{t+1}\right]^{-\gamma-2}\left(\begin{array}{c}\left(\left(1-\widetilde{W}_y\right)^2\sigma_y^2\right) \\ +\left(\widetilde{W}_r^2+w_t^2-2w_t\widetilde{W}_r\right)\sigma_r^2\end{array}\right) \\
-\beta\gamma\mathbb{E}_t\left[c_{t+1}\right]^{-\gamma-1}\left(w_t-\widetilde{W}_r\right)\sigma_r^2
\end{array}\right], \\
c_t + w_t = y_t + r_r w_{t-1}, \qquad (37)
\end{gathered}
$$

$$
\begin{aligned}
y_t &= \left(1-\rho_y\right)\overline{y} + \rho_y y_{t-1} + \varepsilon_{y,t}, \quad \varepsilon_{y,t} \sim N\left(0, \sigma_y^2\right), && (38) \\
r_t &= \left(1-\rho_r\right)\overline{r} + \rho_r r_{t-1} + \varepsilon_{r,t}, \quad \varepsilon_{r,t} \sim N\left(0, \sigma_r^2\right). && (39)
\end{aligned}
$$

For given values of $\widetilde{W}_r$ and $\widetilde{W}_y$, this system of equations has a steady state and regular perturbation techniques can be used to calculate $\widetilde{w}$, $W_r$, and $W_y$. The values for $W_r$ and $W_y$ can then be used to update equation (36). This process is continued until convergence.

# D Details of the ET procedure

In this section, we give detailed information about the implementation of the ET algorithm. The discussion follows the Matlab program closely.[30] To increase transparency, we replace some of the names used in the program with more informative mathematical symbols.[31]

[30] The program is availabe at www.wouterdenhaan.com/software.htm.

[31] In particular, `lhs` in the code stands for $L$, i.e., the left-hand side of the Euler equation; `wlag` in the code stands for wealth chosen previous period, that is, $w_{-1}$; `wconstm` stands for $\phi_0$; `rm` stands for $\widetilde{r}$; `wm` stands

**Define key variables and preliminaries**

- The Gauss-Hermite nodes and weights (scaled by $\sqrt{\pi}$) are stored in the $(N \times 1)$ vectors $X$ and $P$, respectively. We assume that the number of nodes is the same for both random variables.

- The following list of variables are declared to be symbolic variables: $w$, $r$, $w_{-1}$, $y$, $\widetilde{r}$, $\widetilde{w}$, $\widetilde{y}$, $\phi_0$, $\phi_r$, $\phi_w$, $\phi_y$, $\phi_{rr}$, $\phi_{ww}$, $\phi_{yy}$, $\phi_{rw}$, $\phi_{ry}$, $\phi_{wy}$. The last six are only used when implementing second-order ET.

- Define next period's realizations of the two random variables, that is, $r_{+1}$ and $y_{+1}$, as functions of the possible outcomes, i.e., the Gauss-Hermite nodes:

$$\begin{aligned} r_{+1} &= @\left(X\right) \;\; r_{+1} = \left(1-\rho_r\right)\overline{r} + \rho_r r + \sqrt{2}\sigma_r X, && (40) \\ y_{+1} &= @\left(X\right) \;\; y_{+1} = \left(1-\rho_y\right)\widetilde{y} + \rho_y y + \sqrt{2}\sigma_y X. && (41) \end{aligned}$$

  Note that $r_{+1}$ and $y_{+1}$ are not just functions of $X$, but also functions of $r$ and $y$.

- Define the left-hand side of the Euler equation, that is, $c^{-\gamma}$:

$$L = \left(y + r w_{-1} - w\right)^{-\gamma}. \qquad (42)$$

- The key variable is the Euler equation error, which is equal to

$$\begin{gathered} Q_0\left(w, w_{-1}, r, y; \Phi\right) \\ = \qquad\qquad (43) \\ -1 + \frac{\beta}{L}\sum_{i=1}^{N}\sum_{j=1}^{N} r_{+1}\left(X\right)\begin{bmatrix} y_{+1}\left(X\left(j\right)\right) + r_{+1}\left(X\left(i\right)\right)w \\ -\widetilde{h}\left(r_{+1}, w, y_{+1}; \Phi\right) \end{bmatrix}^{-\gamma} P\left(i\right)P\left(j\right), \end{gathered}$$

  where $\Phi$ contains all the coefficients of the approximating functional form. In the program, $Q\left(\cdot\right)$ is calculated using a double "for loop".

for $\widetilde{w}$; `ym` stands for $\widetilde{y}$; `phi_i` stands for $\phi_i$ with `i` $\in${`r,w,y`}; and `phi_ij` stands for $\phi_{i,j}$ with `i,j` $\in${`r,w,y`}; `eq` stands for $Q\left(\cdot\right)$, the Euler-equation error term; `eq_x` stands for $\partial Q/\partial x$ with `x` $\in${`W,r,w,y`}; eq_xz stands for $\partial Q^2/\partial^2 xz$ with `x,z` $\in${`W,r,w,y`}. In both the first and second order derivatives, the modifier $w$ stands for previous period wealth, $w_{-1}$, and the modifier $W$ stands for this period wealth, $w$.

**First-order ET.** When using first-order ET, the function $\widetilde{h}(\cdot)$ is given by

$$\widetilde{h}(r_{+1}, w, y_{+1}; \Phi) = \phi_0 + \phi_r (r_{+1}(X) - \widetilde{r}) + \phi_w (w - \widetilde{w}) + \phi_y (y_{+1}(X) - \widetilde{y}) \tag{44}$$

where

$$\Phi = [\phi_0, \phi_r, \phi_w, \phi_y, \widetilde{r}, \widetilde{w}, \widetilde{y}]. \tag{45}$$

A key aspect of ET is that all elements of $\Phi$ depend on the current values of $r$, $w_{-1}$, and $y$.

The model says that the error term defined in equation (43) should be equal to zero for all $r$, $w_{-1}$, and $y$, that is,

$$Q_0(w, w_{-1}, r, y; \Phi) = 0. \tag{46}$$

Differentiating this equation with respect to $r$, $w_{-1}$, and $y$ gives

$$\begin{aligned} \frac{\partial Q_0}{\partial w}\frac{\partial w}{\partial r} + \frac{\partial Q_0}{\partial r} &= 0, && (47a) \\ \frac{\partial Q_0}{\partial w}\frac{\partial w}{\partial w_{-1}} + \frac{\partial Q_0}{\partial w_{-1}} &= 0, && (47b) \\ \frac{\partial Q_0}{\partial w}\frac{\partial w}{\partial y} + \frac{\partial Q_0}{\partial y} &= 0, && (47c) \end{aligned}$$

With a symbolic toolbox, one defines $Q_0$ as a function, exactly as it is defined in equation (43). Next, one obtains the partial derivatives as functions of $w$, $w_{-1}$, $r$, and $y$ by using the symbolic toolbox' differentiation command.[32] *After* the derivatives have been calculated, one would impose that

$$\begin{aligned} \phi_0 &= w, && (48) \\ \widetilde{r} &= r, && (49) \\ \widetilde{w} &= w_{-1}, && (50) \\ \widetilde{y} &= y. && (51) \end{aligned}$$

That is, the Taylor series expansion to characterize next period's choice is around today's state variables and the function value at those outcomes is, of course, equal to this

[32] In Matlab, `Q_r=diff(Q_0,r)`generates the derivative of `Q_0` with respect to `r`.

period's choice. The derivatives of this Taylor series expansion are equal to the derivatives of this period's choice. That is,

$$\phi_r = \frac{\partial w}{\partial r}, \tag{52a}$$

$$\phi_w = \frac{\partial w}{\partial w_{-1}}. \tag{52b}$$

$$\phi_y = \frac{\partial w}{\partial y}. \tag{52c}$$

After these substitutions, we get the following system of four equations in the four unknowns – $w$, $\phi_r$, $\phi_w$, and $\phi_y$ – for given values of $r$, $w_{-1}$, and $y$:[33]

$$0 = Q_0\left(w, w_{-1}, r, y; \widetilde{\Phi}\right), \tag{53a}$$

$$0 = Q_r\left(w, w_{-1}, r, y; \widetilde{\Phi}\right), \tag{53b}$$

$$0 = Q_w\left(w, w_{-1}, r, y; \widetilde{\Phi}\right), \tag{53c}$$

$$0 = Q_y\left(w, w_{-1}, r, y; \widetilde{\Phi}\right), \tag{53d}$$

where

$$\widetilde{\Phi} = \left[\phi_r, \phi_w, \phi_y\right]. \tag{54}$$

**Second-order ET.** When using second-order ET, the function $\widetilde{h}\left(\cdot\right)$ is given by

$$\begin{gathered} \widetilde{h}\left(r_{+1}, w, y_{+1}; \Phi\right) \\ = \\ \phi_0 + \phi_r\left(r_{+1}\left(X\right) - \widetilde{r}\right) + \phi_w\left(w - \widetilde{w}\right) + \phi_y\left(y_{+1}\left(X\right) - \widetilde{y}\right) \\ +\tfrac{1}{2}\phi_{rr}\left(r_{+1}\left(X\right) - \widetilde{r}\right)^2 + \tfrac{1}{2}\phi_{ww}\left(w - \widetilde{w}\right)^2 + \tfrac{1}{2}\phi_{yy}\left(y_{+1}\left(X\right) - \widetilde{y}\right)^2 \\ +\phi_{rw}\left(r_{+1}\left(X\right) - \widetilde{r}\right)\left(w - \widetilde{w}\right) + \phi_{ry}\left(r_{+1}\left(X\right) - \widetilde{r}\right)\left(y_{+1}\left(X\right) - \widetilde{y}\right) \\ +\phi_{wy}\left(w - \widetilde{w}\right)\left(y_{+1}\left(X\right) - \widetilde{y}\right) \end{gathered} \tag{55}$$

[33] For this system of equations, the nonlinear-equation solver is keen to set $\phi_w$ to values close to $r$ and $\phi_y$ to values close to 1. Since $r > 1$, this implies an explosive solution. It is easy to show that $\phi_w = r$ and $\phi_y = 1$ are exact solutions to equations (53c) and (53d) when $\sigma_r = 0$. To avoid the nonlinear-equation solver choosing or moving towards this possibility, we divide equation (53c) by $r - \phi_w$ and equation (53d) by $1 - \phi_y$.

where

$$\Phi = \left[\phi_0, \phi_r, \phi_w, \phi_y, \phi_{rr}, \phi_{ww}, \phi_{yy}, \phi_{rw}, \phi_{ry}, \phi_{wy}, \widetilde{r}, \widetilde{w}, \widetilde{y}\right]. \tag{56}$$

The additional six equations and associated error terms are given by

$$0 = Q_{ij}\left(w, w_{-1}, r, y; \Phi\right) \quad i, j \in \{r, w, y\} \tag{57}$$

$$Q_{rr}\left(w, w_{-1}, r, y; \Phi\right) = \frac{\partial Q}{\partial w}\phi_{rr} + 2\frac{\partial^2 Q}{\partial w \partial r}\phi_r + \frac{\partial^2 Q}{\partial w^2}\phi_r^2 + \frac{\partial^2 Q}{\partial r^2}, \tag{58}$$

$$Q_{ww}\left(w, w_{-1}, r, y; \Phi\right) = \frac{\partial Q}{\partial w}\phi_{ww} + 2\frac{\partial^2 Q}{\partial w \partial w_{-1}}\phi_w + \frac{\partial^2 Q}{\partial w^2}\phi_w^2 + \frac{\partial^2 Q}{\partial w_{-1}^2}, \tag{59}$$

$$Q_{yy}\left(w, w_{-1}, r, y; \Phi\right) = \frac{\partial Q}{\partial w}\phi_{yy} + 2\frac{\partial^2 Q}{\partial w \partial y}\phi_y + \frac{\partial^2 Q}{\partial w^2}\phi_y^2 + \frac{\partial^2 Q}{\partial y^2}, \tag{60}$$

$$Q_{rw}\left(w, w_{-1}, r, y; \Phi\right) = \left[\begin{array}{c} \frac{\partial Q}{\partial w}\phi_{rw} + \frac{\partial^2 Q}{\partial w \partial w_{-1}}\phi_r + \frac{\partial^2 Q}{\partial w \partial r}\phi_w \\ +\frac{\partial^2 Q}{\partial w^2}\phi_r\phi_w + \frac{\partial^2 Q}{\partial r \partial w_{-1}} \end{array}\right], \tag{61}$$

$$Q_{ry}\left(w, w_{-1}, r, y; \Phi\right) = \left[\begin{array}{c} \frac{\partial Q}{\partial w}\phi_{ry} + \frac{\partial^2 Q}{\partial w \partial y}\phi_r + \frac{\partial^2 Q}{\partial w \partial r}\phi_y \\ +\frac{\partial^2 Q}{\partial w^2}\phi_r\phi_y + \frac{\partial^2 Q}{\partial r \partial y} \end{array}\right], \tag{62}$$

$$Q_{wy}\left(w, w_{-1}, r, y; \Phi\right) = \left[\begin{array}{c} \frac{\partial Q}{\partial w}\phi_{wy} + \frac{\partial^2 Q}{\partial w \partial y}\phi_w + \frac{\partial^2 Q}{\partial w \partial w_{-1}}\phi_y \\ +\frac{\partial^2 Q}{\partial w^2}\phi_w\phi_y + \frac{\partial^2 Q}{\partial w_{-1} \partial y} \end{array}\right]. \tag{63}$$

These equations are obtained by differentiating equation (43) twice and imposing that

$$\begin{aligned} \phi_{rr} &= \frac{\partial^2 w}{\partial r^2}, \phi_{ww} = \frac{\partial^2 w}{\partial w_{-1}^2}, \phi_{yy} = \frac{\partial^2 w}{\partial y^2}, \\ \phi_{rw} &= \frac{\partial^2 w}{\partial r \partial w_{-1}}, \phi_{ry} = \frac{\partial^2 w}{\partial r \partial y}, \phi_{wy} = \frac{\partial^2 w}{\partial w_{-1} \partial y}. \end{aligned} \tag{64}$$

# E Results for other parameter values

The analysis in the main text is based on values for the standard deviations of the innovations that are lower than the ones used in Coeurdacier, Rey, and Winant (2011). The reason for our choice of parameter values is that the model generates very complex nonlinear "escape" dynamics at the original CRW parameter values. More precisely, we find that the solution for wealth generated by an accurate projections method hits the imposed upper bound on wealth when the sample is long enough even when that upper bound is extremely high.[34] This is documented in figure 6, which plots parts of the projection and CRW time paths at the original CRW parameter values. The figure has three panels, each corresponding with a different upper bound on wealth imposed by the projection algorithm. The three values considered are 35, 100, and 1000. The figure documents that increasing the upper bound is not effective in making it not binding. In fact, if the upper bound is increased from 100 to 1000 it actually becomes binding more often in this part of the state space.

Escape dynamics occur at high values of the rate of return, but it is not due to a few extreme values for the rate of return, since the same pattern of results is found when the realizations of $r_t$ are constrained not to be further away from their mean than two standard deviations. As documented in figure 6, the CRW solution does not capture these escape dynamics. The ET algorithm also has difficulty obtaining solutions at many points in the state space.

This type of behavior indicates that the model may not have a stationary solution at those parameter values unless one does impose an upper bound.[35] Projection methods can incorporate such a bound, but this is not possible for CRW and ET. Therefore, we prefer not to do a formal comparison of different solution methods at parameter values where we observe this type of escape dynamics.

# F The neoclassical growth model

In this section, we make clear how easy it is to program the ET algorithm by providing the Matlab code that uses first-order ET to solve the neoclassical growth model, a model that most readers will be familiar with. The program uses the Matlab symbolic toolbox to calculate derivatives. The user also needs a subroutine to generate the Gauss-

[34] At higher values for $\sigma_r$ and $\sigma_y$, it is important for the stability of the projection algorithm that an upper bound is imposed.

[35] It is possible that there is a stationary solution despite the escape dynamics, but that the upper bounds considered here are still not high enough. Imposing the upper bound flattens the policy function for wealth at high interest rates even at values of wealth which are not that close to the upper bound, which makes it even more surprising that the upper bound is reached during simulations.

Hermite quadrature nodes and weights. These are readily available and we provide one at www.wouterdenhaan.com/software.htm. This website also contains the code displayed here and the program to solve the same model with second-order ET.

```
clear;
% This program solves the stochastic Ramsey growth model using the
% ET-algorithm by Den Haan, Kobielarz, and Rendahl (2015).
% Declare parameters for the model

alpha = 1/3;         % Capital share of income:  1/3.
beta = 1.04^(-1/4);    % Discount factor:  4% annual interest rate.
delta = 0.025;     % Depreciation rate.
gamma = 10;     % Coefficient of relative risk aversion.
sigma = 0.03;     % Standard deviation of productivity shock.
rho = 0.95;     % Persistence of technology shock.

% Implied steady state

kss = ((1/beta+delta-1)/alpha)^(1/(alpha-1));
css = kss^(alpha)-delta*kss;

% Information regarding the shock

N_quad = 5;     % Number of nodes for the quadrature (5 goes a long way).
[Z W] = hernodes(N_quad);     % Generate nodes and weights.
W = pi^(-1/2)*W;     % Normalize weights.
Z = Z*sqrt(2)*sigma;     % Normalize nodes.

% Declar symbolic variables

syms k c z phi_k phi_z kt ct zt

kp = exp(z)*k^(alpha)+(1-delta)*k-c;
zp = rho*z+Z;
```

```
% Define the Euler Equation

EE = c-(W’*( beta*(1+exp(zp)*alpha*kp^(alpha-1)-delta)    ...
    .*(ct+phi_k*(kp-kt)+phi_z*(zp-zt)).^(-gamma) ))^(-1/gamma);

% Take derivatives

dk = diff(EE,k);
dz = diff(EE,z);
dc = diff(EE,c);

% The implicit function theorem reveals that
% the following equations should hold:

Ek = phi_k+dk./dc;
Ez = phi_z+dz./dc;

% The collection of equations to be solved is therefore

E = [EE;Ek;Ez];
E = subs(E,[kt zt ct],[k z c]);

% Convert into a matlab function

E = matlabFunction(E,’vars’,{[c;phi_k;phi_z],k,z});

% Solve the problem along a stochastic simulation.

T = 500;
k0 = zeros(1,T+1);
z0 = zeros(1,T+1);
k0(1) = kss;
z0(1) = 0;
% Initial guess
```

```
X = [css;0.9;0.9];
options = optimset('Display','off','TolFun',1e-14,'TolX',1e-14);
rng(20150215,'twister');
e = sigma*randn(T,1);

for t = 1:T
 disp(t)
 E1 = @(x) E(x,k0(t),z0(t));
 X(:,t+1) = fsolve(E1,X(:,t),options);
 k0(t+1) = exp(z0(t))*k0(t)^(alpha)+(1-delta)*k0(t)-X(1,t+1);
 z0(t+1) = rho*z0(t)+e(t);
end
c0 = X(1,:);
% Done.
```

## References

Chamberlain, G., and C. Wilson (2000): "Optimal Intertemporal Consumption Under Uncertainty," *Review of Economic Dynamics*, 3, 365–395.

Coeurdacier, N., H. Rey, and P. Winant (2011): "The Risky Steady State," *American Economic Review: Papers and Proceedings*, 101, 398–401.

Den Haan, W. J. (1995): "The Term Structure of Interest Rates in Real and Monetary Economies," *Journal of Economic Dynamics and Control*, 19, 909–940.

——— (2010): "Comparison of Solutions to the Incomplete Markets Model with Aggregate Uncertainty," *Journal of Economic Dynamics and Control*, 34, 4–27.

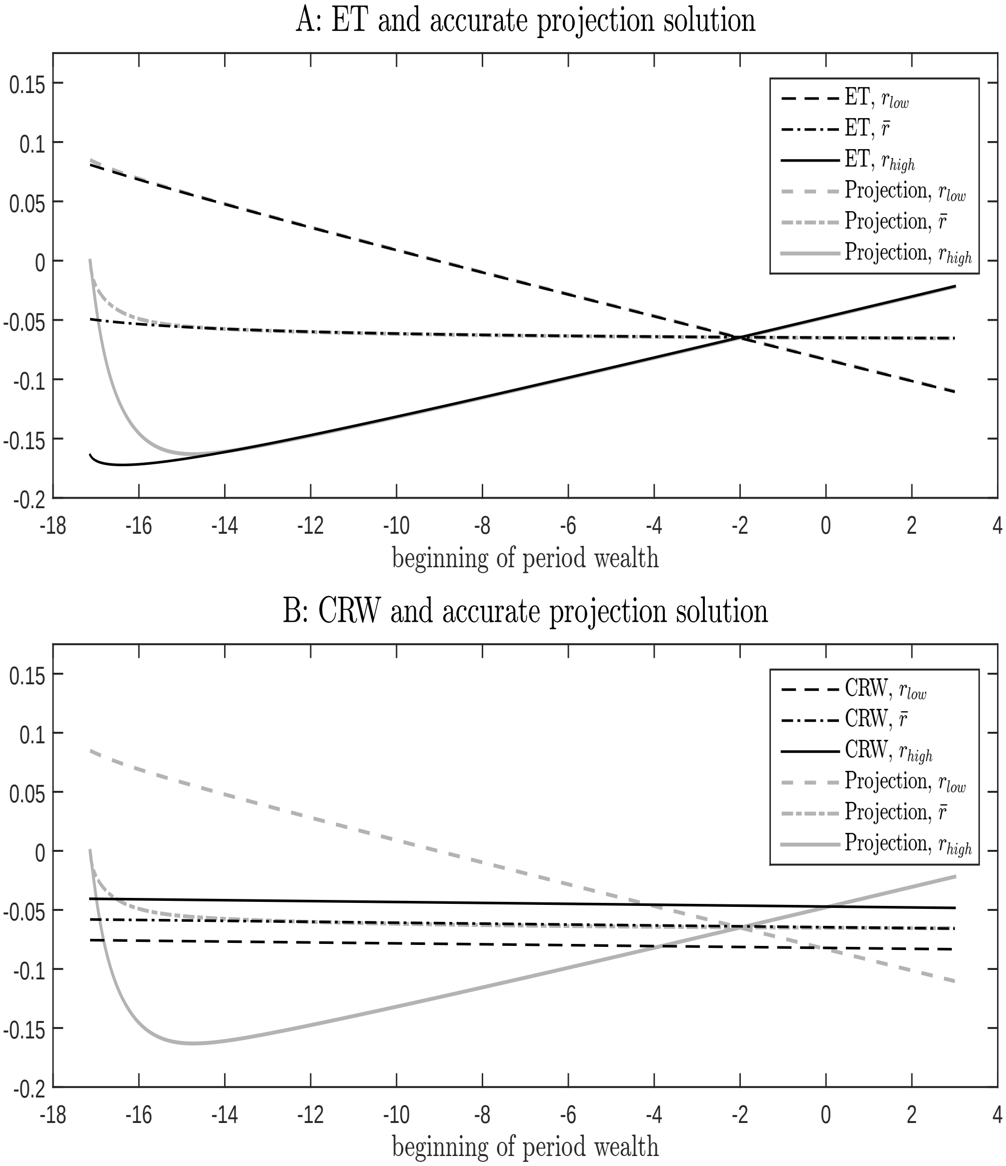
Den Haan, W. J., and J. De Wind (2012): "Nonlinear and Stable Perturbation-based Approximations," *Journal of Economic Dynamics and Control*, 36, 1477–1797.

Den Haan, W. J., and A. Marcet (1990): "Solving the Stochastic Growth Model by Parameterizing Expectations," *Journal of Business and Economic Statistics*, 8, 31–34.

Judd, K. J., S. Maliar, and L. Maliar (2011): "Numerically Stable and Accurate Stochastic Simulation Approaches for Solving Dynamic Economic Models," *Quantitative Economics*, 2, 173–210.

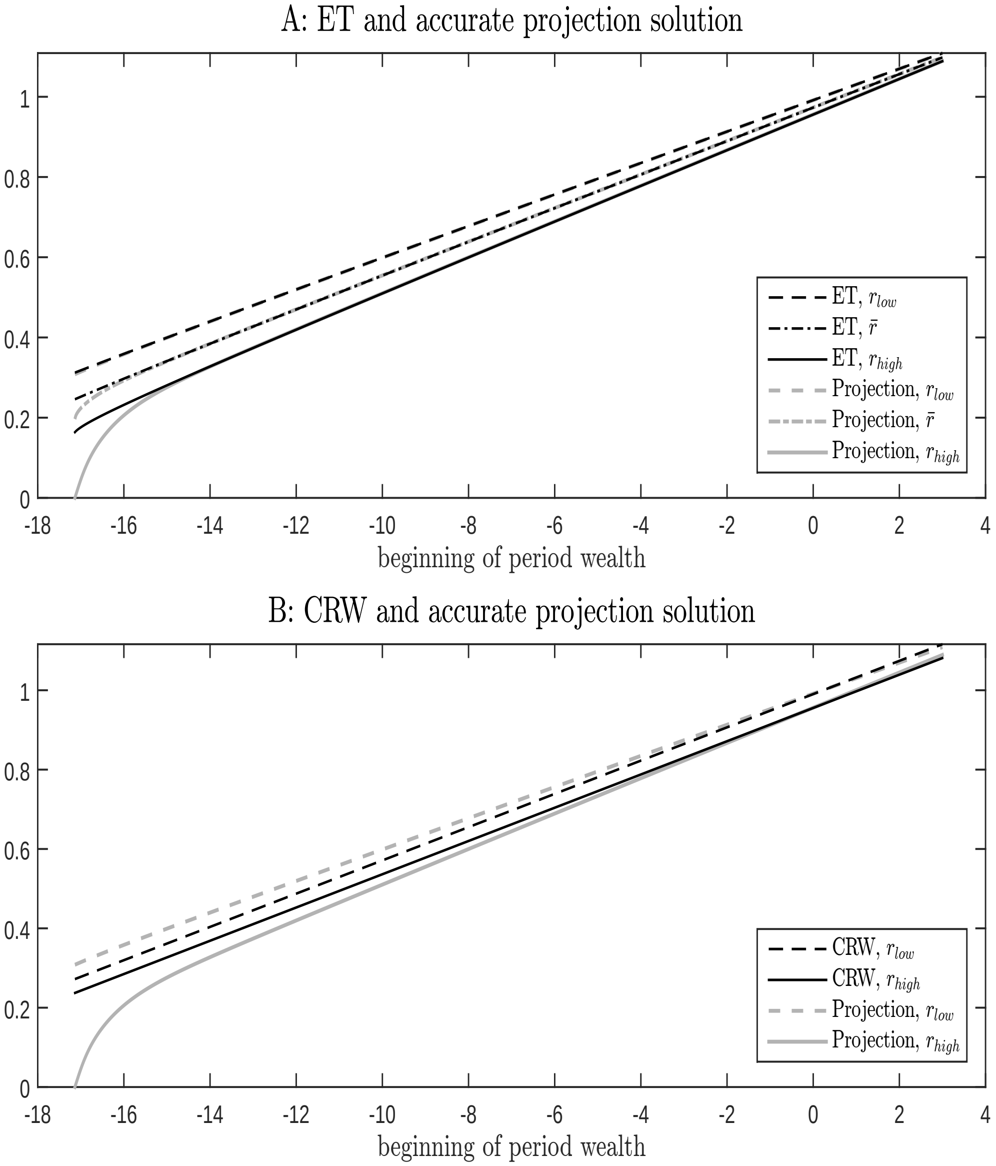
Judd, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58, 410–452.

Figure 1: Wealth Policy Function (low $y$ value)



Notes: These graphs plots the values of the change in wealth, $w - w_{-1}$, according to the indicated algorithm as a function of beginning-of-period wealth, $w_{-1}$, when income, $y$, is equal to 0.9082. The three values of the realized gross rate of return, $r$, are equal to 1.0301, 10358, and 1.0415.

Figure 2: Consumption Policy Function (low $y$ value)

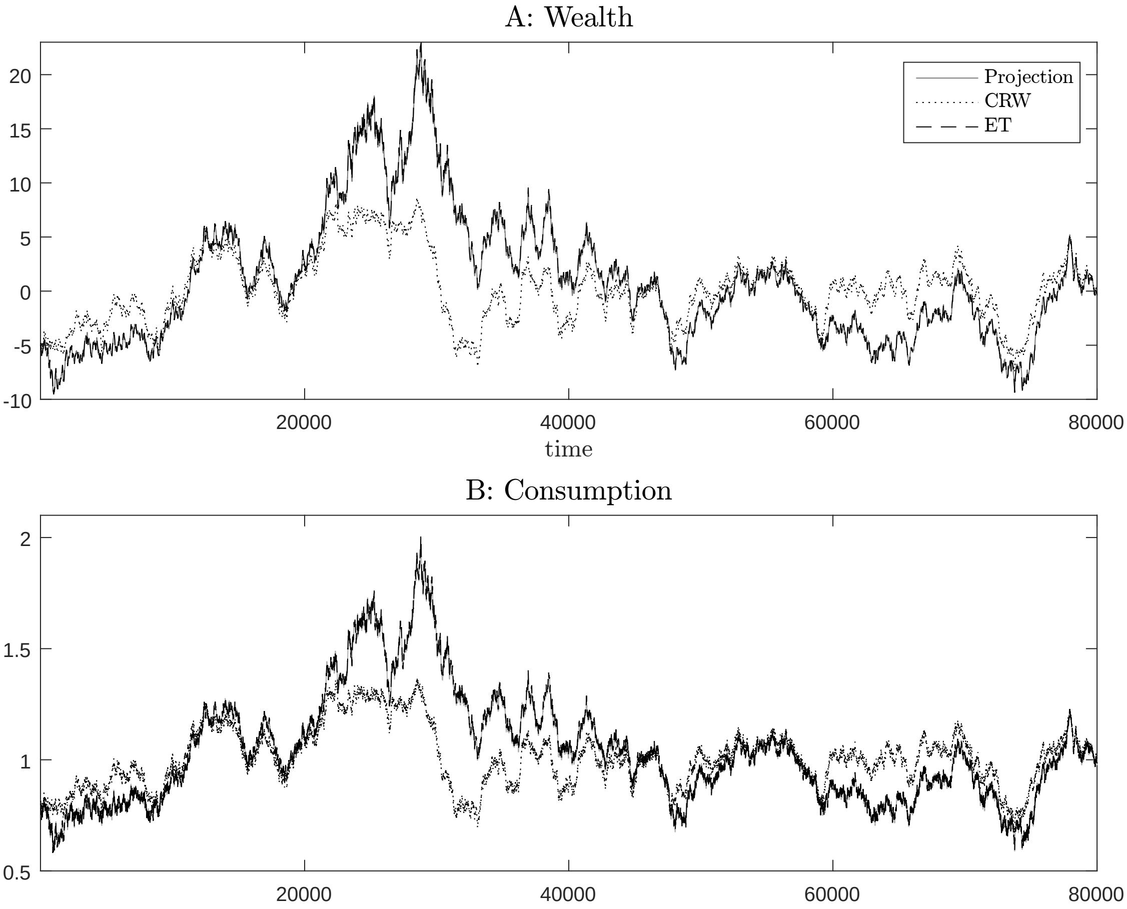


A: ET and accurate projection solution

B: CRW and accurate projection solution

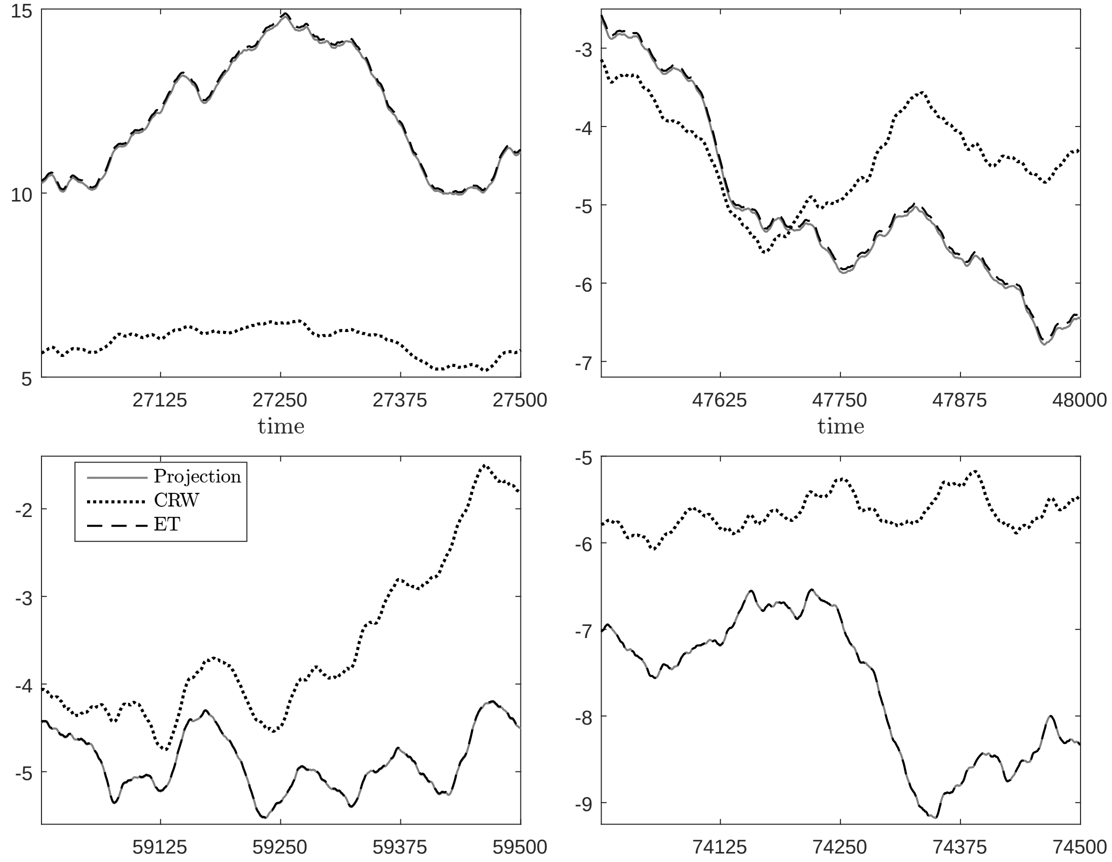
Notes: These graphs plots the values of consumption, $c$, according to the indicated algorithm as a function of beginning-of-period wealth, $w_{-1}$, when income, $y$, is equal to 0.9082. The three values of the realized gross rate of return, $r$, are equal to 1.0301, 10358, and 1.0415.

Figure 3: Simulated Time Paths (whole sample)



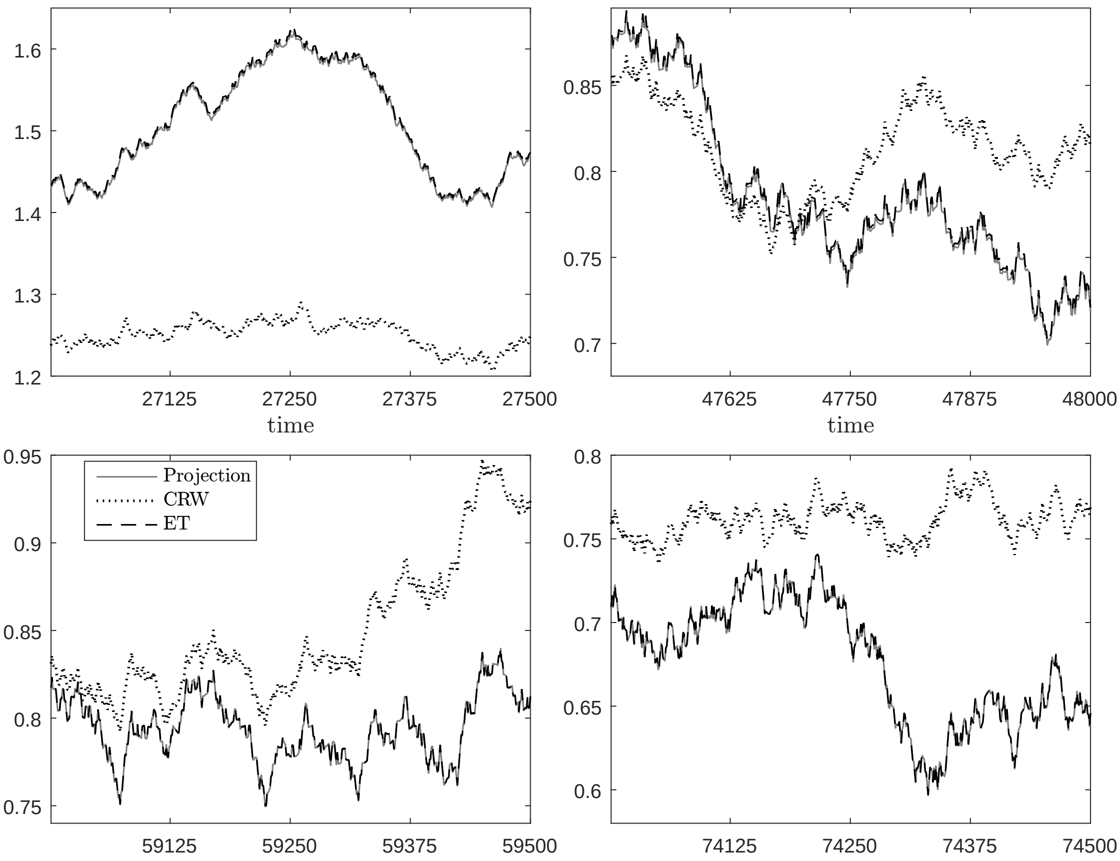
Notes: This graph plots simulated time series generated with the three algorithms using the exact same initial conditions and same draws for the exogenous random variables. Whereas the projection and the ET solution are difficult to distinguish, the CRW solution is clearly different.

Figure 4: Selected Short Samples: Wealth



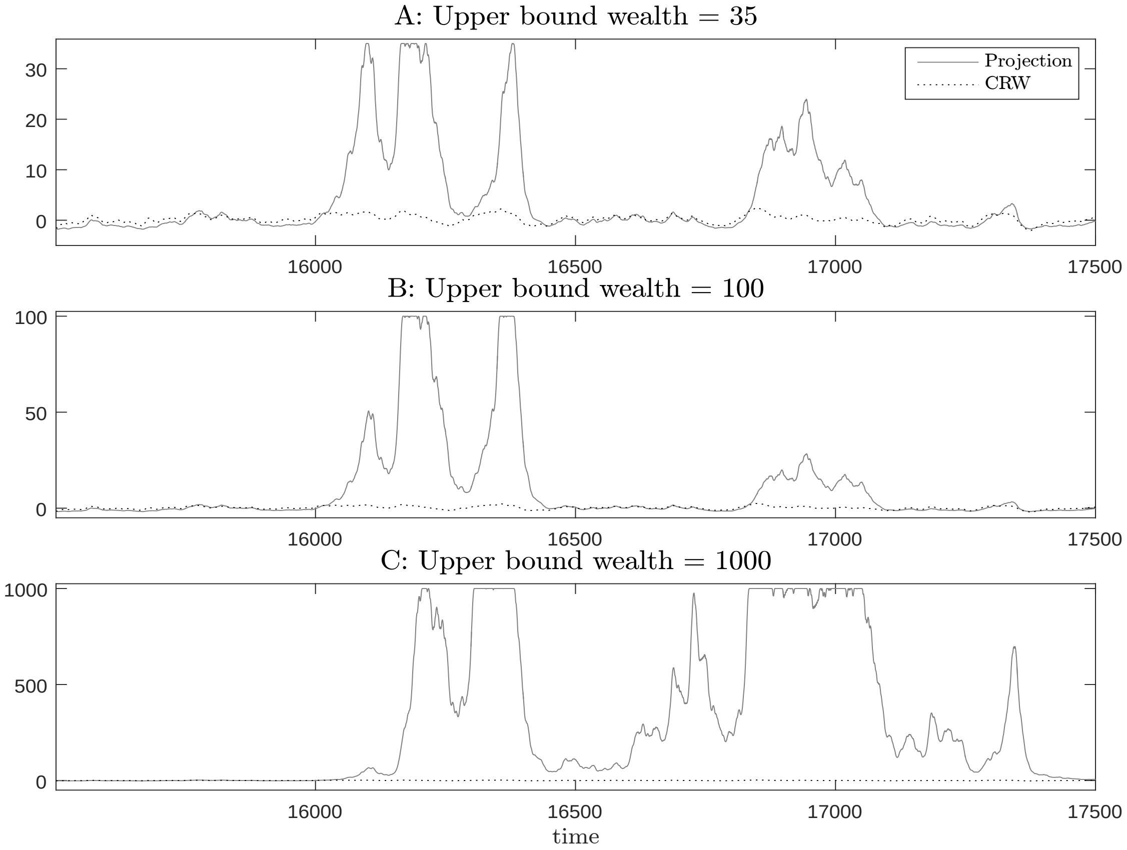
Notes: This graph plots parts of the simulated time series for wealth generated with the three algorithms using the exact same initial conditions and same draws for the exogenous random variables. Whereas the projection and the ET solution are difficult to distinguish, the CRW solution is clearly different.

Figure 5: Selected Short Samples: Consumption



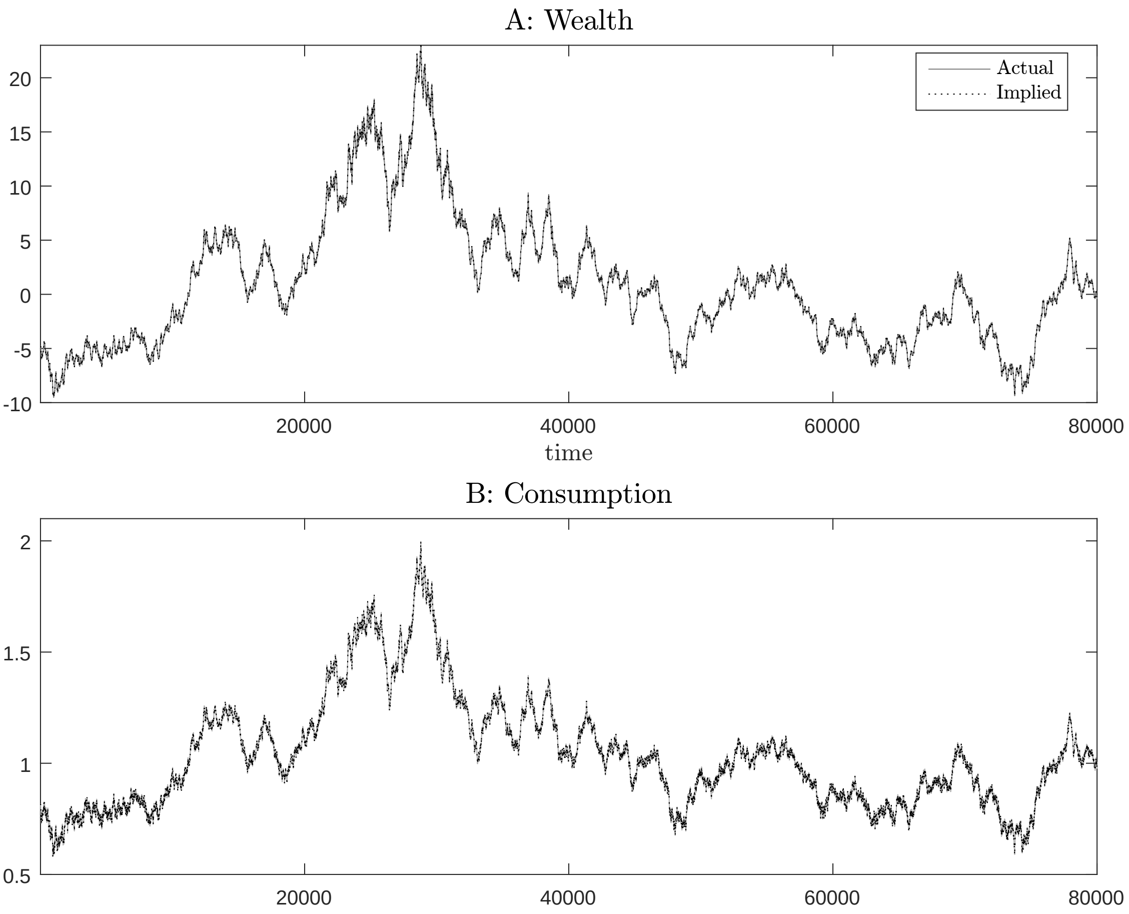
Notes: This graph plots parts of the simulated time series for consumption generated with the three algorithms using the exact same initial conditions and same draws for the exogenous random variables. Whereas the projection and the ET solution are difficult to distinguish, the CRW solution is clearly different.

Figure 6: Simulated Wealth Time Series - Original CRW Parameter Values



Notes: This graph plots simulated time series for wealth generated with the projection and the CRW algorithm using the original CRW parameter values. The projection algorithm imposes the indicated upper bound for wealth.

Figure 7: Dynamic Euler-Equation Accuracy Test



Notes: These graphs plot the actual time series for wealth and consumption and the time series that are the explicit solutions to the first-order conditions using the approximation to calculate next period's choice for wealth. The fact that the two series are difficult to distinguish indicates that the solution is accurate.

Table 1: Parameter Values

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\sigma_r$ | 0.00125 | $\rho_r$ | 0.9 | $\overline{r}$ | 1.04152878685 | $\gamma$ | 4 |
| $\sigma_y$ | 0.01 | $\rho_y$ | 0.9 | $\overline{y}$ | 1 | | |

Table 2: Distance of ET and CRW Outcomes Relative to Accurate Projection Solution

| | ET | CRW |
|---|---|---|
| correlation of $w_t$ with projection solution | 1.000 | 0.729 |
| correlation of $\Delta w_t$ with projection solution | 1.000 | 0.798 |
| correlation of HP-filtered $w_t$ with projection solution | 1.000 | 0.790 |
| mean abs. deviation of $w_t$ from projection as fraction of $SD(w_t)$ | 0.83% | 52.3% |
| max. abs. deviation of $w_t$ from projection as fraction of $SD(w_t)$ | 3.62% | 268.2% |
| correlation of $c_t$ with projection solution | 1.000 | 0.730 |
| correlation of $\Delta c_t$ with projection solution | 1.000 | 0.891 |
| correlation of HP-filtered $c_t$ with projection solution | 1.000 | 0.885 |
| mean abs. deviation of $c_t$ from projection as fraction of $\overline{y}$ | 0.21% | 13.0% |
| max abs. deviation of $c_t$ from projection as fraction of $\overline{y}$ | 0.92% | 68.4% |

Notes: This table reports different measures to indicate the similarity between data generated by the CRW or the ET algorithm and data generated by an accurate projection algorithm. The statistics are based on a simulation of 80,000 observations. $SD(w_t)$ stands for the standard deviation of wealth.

Table 3: Model Properties According to the Three Algorithms: Wealth

| | Projection | ET | CRW |
|---|---|---|---|
| mean of $w_t$ | 0.934 | 0.983 | 0.198 |
| standard deviation of $w_t$ | 5.99 | 6.02 | 3.18 |
| standard deviation of $\Delta w_t$ | 0.022 | 0.022 | 0.017 |
| standard deviation of HP-filtered $w_t$ | 0.041 | 0.041 | 0.031 |
| correlation of $w_t$ and $y_t$ | 0.033 | 0.032 | 0.076 |
| correlation of $w_t$ and $r_t$ | -0.004 | -0.004 | -0.004 |
| correlation of $w_t$ and $c_t$ | 0.999 | 0.999 | 0.998 |
| correlation of $\Delta w_t$ and $\Delta y_t$ | 0.168 | 0.168 | 0.216 |
| correlation of $\Delta w_t$ and $\Delta r_t$ | 0.068 | 0.069 | 0.060 |
| correlation of $\Delta w_t$ and $\Delta c_t$ | 0.375 | 0.376 | 0.327 |
| correlation of HP-filtered $w_t$ and $y_t$ | 0.118 | 0.117 | 0.144 |
| correlation of HP-filtered $w_t$ and $r_t$ | 0.048 | 0.049 | 0.045 |
| correlation of HP-filtered $w_t$ and $c_t$ | 0.432 | 0.433 | 0.370 |

Notes: This table reports model properties for data generated by the indicated algorithm. The results are based on a simulation of 80,000 observations.

Table 4: Model Properties According to the Three Algorithms: Consumption

| | Projection | ET | CRW |
|---|---|---|---|
| mean of $c_t$ | 1.039 | 1.041 | 1.008 |
| standard deviation of $c_t$ | 0.250 | 0.251 | 0.134 |
| standard deviation of $\Delta c_t$ | 0.0038 | 0.0038 | 0.0036 |
| standard deviation of HP-filtered $c_t$ | 0.0050 | 0.0050 | 0.0046 |
| correlation of $c_t$ and $y_t$ | 0.057 | 0.056 | 0.121 |
| correlation of $c_t$ and $r_t$ | -0.020 | -0.020 | -0.037 |
| correlation of $\Delta c_t$ and $\Delta y_t$ | 0.747 | 0.747 | 0.808 |
| correlation of $\Delta c_t$ and $\Delta r_t$ | -0.455 | -0.452 | -0.560 |
| correlation of HP-filtered $c_t$ and $y_t$ | 0.723 | 0.723 | 0.788 |
| correlation of HP-filtered $c_t$ and $r_t$ | -0.441 | -0.439 | -0.546 |

Notes: This table reports model properties for data generated by the indicated algorithm. The results are based on a simulation of 80,000 observations.

Table 5: Accuracy Test

| | mean<br>absolute error | maximum<br>absolute error |
|---|---|---|
| consumption | | |
| as fraction of $\overline{y}$ | $9.2e-5$ | $4.0e-4$ |
| as fraction of $SD(w_t)$ | $1.5e-5$ | $6.7e-5$ |
| wealth | | |
| as fraction of $\overline{y}$ | $2.2e-3$ | $9.4e-3$ |
| as fraction of $SD(w_t)$ | $3.7e-3$ | $1.6e-3$ |

Notes: This table reports the errors of the dynamic Euler equation accuracy test using a simulation of 80,000 observations.