TE TAI ŌHANGA
THE TREASURY

Solving Stochastic OLG Models Using Chebyshev Parameterized Expectations

Murat Özbilgin and Robert Kirkby

New Zealand Treasury Working Paper 24/03

June 2024

Te Kāwanatanga o Aotearoa
New Zealand Government

NZ TREASURY WORKING PAPER 24/03

Solving Stochastic OLG Models Using Chebyshev Parameterized Expectations

MONTH/YEAR

June 2024

AUTHORS

Murat Özbilgin
New Zealand Treasury
No. 1 The Terrace
Wellington
New Zealand
Email: murat.ozbilgin@treasury.govt.nz
Telephone: +64 4 439 6365

Robert Kirkby
Victoria University of Wellington
RH 712, Rutherford House
23 Lambton Quay
Wellington
New Zealand
Email: robert.kirkby@vuw.ac.nz
Telephone: +64 4 463 5893

URL

Treasury website at June 2024:
https://www.treasury.govt.nz/publications/wp/wp-24-03

ACKNOWLEDGEMENTS

We would like to thank Lilia Maliar, Johannes Brumm, Christie Smith, Megan Stephens, and Burkhard Heer for their helpful comments and suggestions. Any remaining errors are ours.

NZ TREASURY

New Zealand Treasury
PO Box 3724
Wellington 6140
NEW ZEALAND
Email: information@treasury.govt.nz
Telephone: +64 4 472 2733
Website: www.treasury.govt.nz

Abstract

This paper presents an efficient solution method for solving stochastic overlapping generations (S-OLG) models. We use the Chebyshev parameterized expectation algorithm (C-PEA) developed by Christiano and Fisher (2000) to solve the life cycle block of S-OLGs. The method is well suited for this family of models, capable of handling nonlinearities inherent in the life-cycle aspect of S-OLGSs, and occasionally binding constraints associated with borrowing constraints. We carefully examine practical considerations and describe how to efficiently implement this method. To illustrate the method's effectiveness, we apply it to solve a standard S-OLG model with idiosyncratic risk and two permanent types. We calculate Euler equation errors throughout the life cycle and measure computational time to demonstrate that C-PEA can perform well under these computational challenges with reasonable accuracy and efficiency. Our results show that, together with its scalability to higher dimensional problems, C-PEA can be a valuable tool for policy analysts and researchers working with S-OLG models.

JEL CLASSIFICATION

C63 Computational Techniques; Simulation Modeling
C68 Computable General Equilibrium Models
E2: Consumption, Saving, Production, Employment, and Investment

KEYWORDS

Chebyshev interpolation; Parameterized expectations; Overlapping generations models

Executive summary

Recently, there has been a growing emphasis on understanding how economic policies impact not only economic efficiency but also the distribution of benefits and costs across society. Similarly, one of the New Zealand Treasury's top priorities is the development and integration of the living standards framework (LSF) and He Ara Waiora into its policy advice. These frameworks center on considering the well-being of different generations when evaluating policies and aim to improve our understanding of how policies affect different groups within society.

Stochastic overlapping generations (S-OLG) models are key tools for analyzing policies that have repercussions for various generations and groups. As part of a broader spectrum of structural models, S-OLG models enable us to conduct counterfactual policy experiments and comprehensive analyses that take into account interactions and responses from various parts of the economy under general equilibrium conditions.

However, even the simplest structural models involve significant mathematical complexity, and even for most basic versions, it is not possible to pin down the equilibrium outcomes and conduct policy analysis with paper and pencil, or spreadsheet methods, without resort to numerical approximation and computational techniques. These complexities are known to be particularly challenging for S-OLG models. While the availability of more comprehensive and detailed micro data has enabled the refinement of these models using real-world information, the primary challenge in using S-OLG models lies in the computational difficulties associated with solving them to reach equilibrium outcomes and conduct policy experiments.

This research aims to contribute to these issues by providing an efficient method for solving complex S-OLG models and support the Treasury's efforts to incorporate the LSF and He Ara Waiora into our policy analysis. This tool will bolster our capacity to assess and enhance macroeconomic policy advice, especially concerning the sustainability of fiscal policy, superannuation, taxes, transfers, demographics, and more.

Currently, one of the work agendas of the Treasury's Modelling and Research Team aims to analyse the intergenerational implications of alternative pension policies. The solution approach developed and documented in this paper constitutes one of the core methods to solve the overlapping generations model built under this research agenda.

In the following sections, we demonstrate the application of Chebyshev polynomials in conjunction with the Parameterized Expectations Algorithm (C-PEA) as a solution method for S-OLG models. We offer a comprehensive guide on implementing C-PEA for these models and present the outcomes of its application to a basic S-OLG model, including details on runtime and accuracy. We also discuss some key decisions that need to be made when using these algorithms. Our findings indicate that C-PEA is a robust solution method that maintains a good balance between accuracy and speed. Furthermore, C-PEA requires fewer parameters than most alternative methods, showing promise for its application to more complex and higher-dimensional problems in the future.

Contents

List of Tables

List of Figures

Solving Stochastic OLG Models Using Chebyshev Parameterized Expectations

1. Introduction

Stochastic overlapping generations models (S-OLGs) are a core tool for economic policy analysis. These models offer a far richer and more realistic framework than those with a representative agent. The recent increased emphasis on the distributional effects of economic policy alongside traditional efficiency concerns has elevated interest in these models. Among the most salient aspects of heterogeneity is intergenerational variation, to which S-OLG models speak directly. The greater availability of microdata has also facilitated a more data-driven calibration of these models. Perhaps the main limiting factors to wider use of S-OLG models are the computational challenges of solving them.

This paper presents an efficient method for solving high-dimensional, policy-oriented S-OLG models. It aligns with New Zealand Treasury's strategic priorities, specifically the development and integration of the living standards framework (LSF) and He Ara Waiora into its policy advice. These frameworks emphasise an intergenerational approach to well-being and seek to enhance our understanding of the distributional impacts of policies. Stochastic OLG models are well-suited for intergenerational policy analysis, and this research supports the Treasury's efforts to incorporate both the LSF and He Ara Waiora in our analysis and macroeconomic policy advice on the sustainability of fiscal policy, superannuation, taxes, transfers, and demographics among others.[1]

We demonstrate the use of Chebyshev polynomials together with the Parameterized Expectations Algorithm, C-PEA for short, as a method to compute the stationary equilibrium of S-OLGs.[2] We provide a comprehensive explanation of how to implement C-PEA for S-OLG models. We then document the run time and accuracy of using C-PEA to solve a simple S-OLG model, and discuss some of the decisions that have to be made when implementing these algorithms. We find that C-PEA is a robust solution method that produces a good mix of accuracy and speed. Further, C-PEA requires much fewer parameters than most methods and thereby offers promise for generalization to higher dimensional problems.

A variety of aspects contribute to the computational challenges of S-OLGs. One is that in S-OLGs the endogenous state displays considerable variation across age cohorts. The life-cycle structure implies that individuals typically start life with zero or a low level of assets and accumulates assets up to a certain point in their life cycle after which the assets are entirely consumed until the end of life (in the absence of a bequest motive). Hence we need to be careful about where accuracy is being concentrated while solving such models; in practice this means we use an expanded Chebyshev grid, which we describe in more depth later. Idiosyncratic shocks also contribute to the dispersion of agents in S-OLGs. S-OLGs also involve an endogenous borrowing constraint, as well as non-negativity constraints for choice variables, which may become binding within the individual state space.

1 For fundamental concepts and mechanisms of a workhorse OLG model, see Binning et al. (2024).

2 The methods presented in this paper can be readily applied to solving transitional dynamics, although our focus in this paper does not include the study of transitions.

This paper uses C-PEA to solve stochastic overlapping generations models. Chebyshev polynomials are used to parameterize the conditional expectations term in the Euler equations, known as the parameterized expectations approach (PEA). Previous applications of Chebyshev polynomials to dynamic general equilibrium models have used them to approximate the policy function (Krueger and Kubler 2004, Malin, Krueger and Kubler 2011, Judd, Maliar, Maliar and Valero 2014) or the value function (Cai, Judd, Lontzek, Michelangeli and Su 2017).[3] But Chebyshev polynomials work better when approximating smooth functions (Judd, 1998), and for the life-cycle problems in OLG models it is well known that periodically-binding constraints can lead to kinks in the policy and value functions. In contrast, the integral underlying the conditional expectation term that we approximate is likely to be smoother, and thus the Chebyshev polynomials can provide a better fit in principle. Our approach builds on two papers: first, Christiano and Fisher (2000), who introduced Chebyshev polynomials and numerical integration into the original parameterized expectations algorithm by den Haan and Marcet (1990), and second, Binder et al. (2000), who introduced the Chebyshev parameterized expectations algorithm to a life cycle model. Our main contribution is to provide guidelines for the implementation of C-PEA for an S-OLG model, along with evidence of its performance.

In the rest of this article we first turn to an explanation of the C-PEA approach to solving S-OLG models, including how to compute the value function as this is not part of the solution itself when using C-PEA. We then describe a simple S-OLG and document the results of applying C-PEA to this model in terms of run times and accuracy. We look at how the order of the Chebyshev polynomials, a decision that has to be made when implementing the code, influences the run times and accuracy.

2. Solving S-OLGs with Chebyshev-PEA

The steady-state general equilibrium solution of a typical S-OLG model follows the steps outlined in Figure 1.

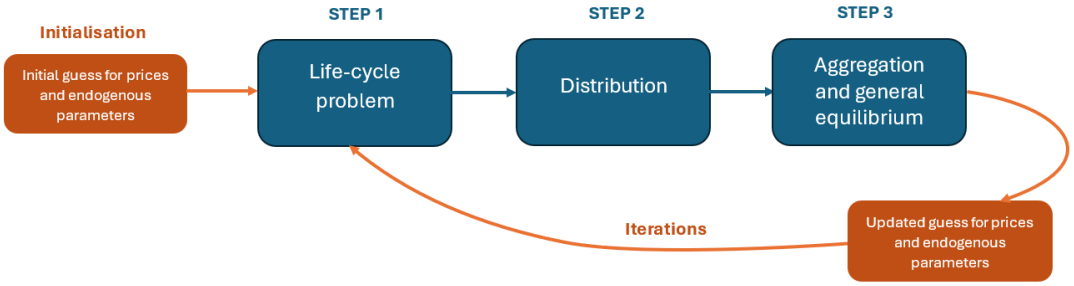
In step 1, the individual decision rules are solved for agents in each generation and each permanent type if the model features ex-ante heterogeneity.

In step 2, the decision rules obtained in step 1 are used to compute the distribution of agents for each permanent type, each generation, and each initial point of idiosyncraticity.

In step 3, these distributions are used to compute aggregate variables, prices, and endogenous parameters (typically taxes or transfers). The model is then resolved with the updated prices and endogenous parameters until they converge to satisfy all aggregate accounting identities such as resource and budget constraints.

[3] On the theory side Munos and Szepesvari (2008) prove that fitted value function iteration using Chebyshev polynomials in combination with random state space sampling will converge as the order of the Chebyshev polynomials goes to infinity.

Figure 1 – Steps for solving an S-OLG model



Binder et al. (2000) advocate using weighted residual methods in life cycle models as an efficient approach that can account for the effects of uncertainty on household decisions. Our approach is similar, as we employ Chebyshev-PEA in step 1 and use Chebyshev polynomials in step 2 of the solution process. Specifically, we use Chebyshev regressions, and time iterations as inner loops of Step 1, which, as we demonstrate later, are very fast and efficient in the finite horizon case. We provide a detailed implementation guide for this method and document its accuracy and computational costs. Our analysis demonstrates that using Chebyshev-PEA in a baseline S-OLG model offers a highly efficient algorithm, showing promise for handling complex S-OLG models effectively.

Next, we explain how the Chebyshev-PEA method can be used to solve individual decision rules in step 1 of the solution process.

2.1 Solving the agents' problem

Many problems in economics admit the recursive representation,

$$\begin{aligned} V(u,z) = &\max_{x} \left\{ U(x,u,z) + \beta \, \mathbb{E}\left[V\left(u',z'\right) \mid u,z \right] \right\} \\ \text{s.t.} \quad & z' = \zeta\left(z\right) + \varepsilon', \\ & u' = \phi\left(x,u,z\right). \end{aligned} \tag{1}$$

Above, $V(\cdot)$ is the value function, which is essentially the optimal value of lifetime utility. $\mathbb{E}$ denotes the expectation operator, aggregating the next period's value under all possible probabilistic outcomes. $x \in X \subseteq \mathbb{R}^{n_x}$ is a set of control variables, $u \in U \subseteq \mathbb{R}^{n_u}$ is a set of endogenous state variables, $z \in Z \subseteq \mathbb{R}^{n_z}$ are exogenous state variables, ε are i.i.d. shocks, β is a discount factor, and primes indicate next period values.

Under certain conditions the problem above yields the following optimality conditions:

$$\frac{\partial U\left(x,u,z\right)}{\partial x} + \frac{\partial \phi\left(x,u,z\right)}{\partial x} \beta \, \mathbb{E}\left[\frac{\partial U\left(x',u',z'\right)}{\partial u'} + \frac{\partial \phi\left(x',u',z'\right)}{\partial u'} \left. \frac{-\frac{\partial U(x',u',z')}{\partial x'}}{\frac{\partial \phi(x',u',z')}{\partial x'}} \right| u,z \right] = 0 \tag{2}$$

Hence, we can characterize the solution to the problem as a system of stochastic difference equations. Denote the expression in the expectation operator above as a function $\psi\left(\cdot\right)$ that maps $X \times U \times Z$ to a set $H \subset \mathbb{R}^{n_x+n_u+n_z}$. Then we can denote the overall system defining the equilibrium by a function Υ given by the equation below:

$$\Upsilon\left[\mathbb{E}\left[\psi\left(x',u',z'\right) \middle| u,z \right], u', x, u, z \right] = 0 \tag{3}$$

The recursive nature of the problem in an infinite horizon setting implies that there is a time-invariant conditional expectation function, $\Psi^{*}\left(u,z\right)$, that solves equation (3).

2.1.1 Parameterized expectations algorithm

The parameterized expectations algorithm (PEA) by den Haan and Marcet (1990) involves approximating the expectational function $\mathbb{E}\left[\psi\left(x', u', z'\right)|u, z\right]$ in (3) with a function $g\left(u, z; \Lambda\right)$. This approximation requires determining the parameters $\hat{\Lambda}$ such that

$$\Upsilon\left[g(u, z; \hat{\Lambda}), u', x, u, z\right] = 0 \tag{4}$$

The PEA employs Monte-Carlo simulations to compute $\hat{\Lambda}$. In the first step of the algorithm, a large number of innovations to z are generated. Subsequently, the model is solved forward from the initial values of u and z (typically the steady state values), using a guess for the parameters, $\hat{\Lambda}^0$. Specifically, the approximating function $g(\cdot\,; \hat{\Lambda}^0)$ yields values for the control variables, x, and the next period's endogenous states, u' through (4).

In the final step, the data generated on $\psi(\,\cdot\,)$ are regressed on $g(\cdot\,; \hat{\Lambda}^0)$ to obtain an updated estimate of Λ, $\hat{\Lambda}^1$. This iterative process continues until convergence is achieved.[4]

As highlighted by Christiano and Fisher (2000), the original PEA of den Haan and Marcet (1990) is a fragile and inefficient algorithm, in terms of convergence and accuracy, due to two underlying problems: First, ordinary polynomials of varying degrees are multicollinear. As a result, higher-order polynomials become impractical to increase the accuracy of the solutions. Second, the simulations lead to the concentration of simulations around highly probable areas of the invariant distribution of the state variables leading to an inferior global solution. Christiano and Fisher (2000) introduce Chebyshev regressions and numerical integration to the original PEA to remedy these problems, which we turn to next.

2.1.2 Chebyshev regressions

The idea behind Chebyshev regressions is similar to ordinary least squares in econometrics. Suppose we want to approximate a function $y = f\left(x\right)$ where $x \in \mathbb{R}^k$, using m data points on x. The least squared approximation of $f(x)$ with functions $\phi_j\left(x\right)$ involves finding parameters $a_1, \cdots, a_n$ that satisfy

$$\min_{\{a_1,\ldots,a_n\}} \sum_{i=1}^{m} \left(a_1\phi_1(x_i) + \cdots + a_n\phi_n(x_i) - y_i\right)^2. \tag{5}$$

If the least squares method is used in a statistical exercise. the points x_i are given by the empirical data set. Also, most of the time we are restricted to a small set of functions due to concerns about degrees of freedom. Consequently, in statistical exercises, we do not have control over the errors, $y_i - \hat{f}(x_i)$.

In numerical applications, on the other hand, the points $x_i, \cdots, x_m$, the functions $\phi(\cdot)$, and the polynomial degree n can all be hand-picked to construct good approximations, as explained in detail in Judd (1998). A specific kind of polynomials, orthogonal polynomials, have particular importance in numerical applications, because any polynomial approximation of a function can be expressed in terms of orthogonal polynomials. Further, orthogonal polynomials can be evaluated fast and efficiently because they obey certain recursion relations.

[4] For further details and an illustrative application, refer to den Haan and Marcet (1990). For a textbook discussion, see also Heer and Maussner (2009).

Chebyshev polynomials are a special family of orthogonal polynomials and have particularly desirable properties for numerical approximation, as described in two theorems reported in Judd (1998).[5] The first theorem states that the least squares approximation of a smooth function with Chebyshev polynomials converges uniformly when the polynomial degree is increased. The second theorem establishes that, for such an approximation, the Chebyshev coefficients must drop off rapidly. Hence, the quality of a solution can be judged by inspecting the Chebyshev coefficients. Furthermore, Chebyshev least squares approximation has the near-minimax property,[6] which implies that Chebyshev approximations provide near best polynomial approximations in L^{∞} norms as well.

In practice, least squares approximations require integrals that cannot be evaluated exactly. They can be approximated at a finite set of points but that can be costly. Therefore, Chebyshev polynomials are mostly used as interpolating polynomials in practical applications. Judd (1998) provides two further theorems,[7] which establish that interpolation of smooth functions with Chebyshev polynomials at zeros of these polynomials attain the minimum possible upper bound for the interpolation error.

Chebyshev regressions combine Chebyshev least squares approximation and Chebyshev interpolation. The algorithm we use in this study is described below, and is essentially the same as Algorithm 6.2 in Judd (1998).[8]

Definition: *Chebyshev regression algorithm*

Purpose: Approximate function $f(x)$ with Chebyshev polynomials:

$$\hat{f}(x) = \sum_{k=0}^{n} \lambda_k \varphi_k(x),$$

where $\varphi_k(\cdot)$ is Chebyshev polynomial of degree k.

1. Choose the number of interpolation nodes, m, and the polynomial degree, n.
2. Create interpolation nodes, z, which are the zeros of degree m Chebyshev polynomials.
3. Transform z from the Chebyshev domain, $[-1, 1]$, into $[a, b]$ using the formula

$$\bar{z} = \frac{z+1}{2}(b-a) + a.$$

4. Create an $m \times n$ regression matrix Ψ. The rows are Chebyshev polynomials of degree 1 to n evaluated at z.
5. Get $n \times 1$ coefficients Λ:

$$\Lambda = \left(\Psi' \Psi\right)^{-1} \Psi' f(\bar{z}).$$

2.1.3 Chebyshev parameterized expectations algorithm (C-PEA)

C-PEA is a modification of the original PEA method introduced by den Haan and Marcet (1990). C-PEA replaces the nonlinear regression step in PEA with Chebyshev regressions,

[5] See Theorems 6.4.2 and 6.4.3 in Judd (1998).
[6] See Theorem 6.5.4 in Judd (1998).
[7] See Theorems 6.7.1 and 6.7.2 in Judd (1998).
[8] For information on common families of orthogonal polynomials and their recursion formulas, refer to Section 6.3 in Judd (1998).

which are known for their ability to approximate smooth functions accurately. This modification was proposed by Christiano and Fisher (2000), who recognized the limitations of the original PEA and the advantages of using Chebyshev polynomials.

The use of Chebyshev polynomials in C-PEA provides several benefits. First, Chebyshev polynomials are more powerful in approximating smooth functions compared to other methods, such as polynomial or spline approximations. This makes them well-suited for approximating expectational functions in economic problems, which are often smooth. In contrast, policy functions may have kinks or discontinuities, making them harder to approximate accurately using traditional methods.

Second, C-PEA improves the accuracy and efficiency of approximating expectations compared to Monte Carlo simulations used in the original PEA. Monte Carlo simulations involve random simulations and averaging the outcomes, which can be computationally inefficient because the simulations have a mass on frequently visited areas of the state space. In contrast, C-PEA uses numerical integration and Chebyshev regressions, which provide more precise and efficient approximations of expectational functions. Chebyshev polynomials used as regressors in C-PEA have the advantage of orthogonality, which means that each successive, higher degree polynomial represents an independent source of information without duplicating information already captured by lower order polynomials. This leads to a more efficient representation of the underlying function being approximated compared to other regression methods.

Going back to the problem in (3), the C-PEA amounts to approximating $E\left[\psi\left(x', u', z'\right)|u, z\right]$ using Chebyshev polynomials as basis functions and the zeros of Chebyshev polynomials as the interpolation nodes. Let us assume that u and z are one-dimensional for expositional brevity. The C-PEA involves the following steps:

1. Choose a degree of Chebyshev polynomials for each state variable: p_u and p_z.
2. Choose a number of interpolation points for each state variable: m_u and m_z.
3. Choose upper and lower limits for u and z that will be used in the approximation process.
4. Construct the interpolation nodes, $\bar{u}$ and $\bar{z}$.
   - Use m_u zeros of the Chebyshev polynomials of degree m_u, say r_u for u and m_z zeros of Chebyshev polynomials of degree m_z, say r_z for z.
   - The nodes for u are $\bar{u} = \tau\left(r_u\right)$ and the nodes for z are $\bar{z} = \tau\left(r_z\right)$, where the operator $\tau\left(\cdot\right)$ transforms Chebyshev zeros to the domain of the model variables.
5. Construct the function, $g\left(u, z; \Lambda\right)$ to approximate $\mathbb{E}\left[\psi\left(x', u', z'\right) \mid u, z\right]$.
   (a) Choose a starting value for Λ, Λ^0 with elements λ_{ij}^0, $i = 1, ..., p_u$, and $j = 1, ..., p_z$.
   (b) Set the approximating function as:

$$g\left(u, z; \Lambda^0\right) = \sum_{i=0}^{p_u} \sum_{j=0}^{p_z} \lambda_{ij}^0 T_i\left[\tau^{-1}\left(\bar{u}\right)\right] T_j\left[\tau^{-1}\left(\bar{z}\right)\right].$$

   where $T_i\left(\cdot\right)$ denotes the Chebyshev polynomial of degree i.
6. Require $g\left(u, z; \Lambda^0\right)$ to equal to $\mathbb{E}\left[\psi\left(x', u', z'\right) \mid u, z\right]$ exactly at collocation points. In other words,

$$\Upsilon\left[g\left(u, z; \Lambda^0\right), u', x, u, z\right] = 0$$

7. Solve for the control variables x and the endogenous state variables u' using the condition above.

8. Let the next period's value for the exogenous state be z'.

   (a) Construct $g\left(u', z'; \Lambda^0\right)$. Require that:

   $$\Upsilon\left[g\left(u', z'; \Lambda^0\right), u'', x', u', z'\right] = 0$$

   (b) Solve for x' using the condition above.

9. Calculate the implied value of $\mathbb{E}\left[\psi\left(x', u', z'\right) \mid u, z\right]$ under Λ^0, denoted $\Phi\left(\Lambda^0\right)$. Note that

   $$\Phi\left(\Lambda^0\right) = \int_{-\infty}^{\infty} \psi\left(x', u', z'\right) f\left(z' \mid z\right) dz.$$

10. Run an OLS regression of $\Phi\left(\Lambda^0\right)$ on $\sum_{i=0}^{p_u} \sum_{j=0}^{p_z} T_i\left[\tau^{-1}\left(\bar{u}\right)\right] T_j\left[\tau^{-1}\left(\bar{z}\right)\right]$ to get the guess for the updated vector of parameters, Λ^1.

11. If $\left\|\Lambda^1 - \Lambda^0\right\| < \epsilon$, for ϵ being very small, stop and set $\Lambda^* = \Lambda^1$. Then we have the solution for the expectational function, $\Psi^*\left(u, z\right) = g\left(u, z; \Lambda^*\right)$. If $\left\|\Lambda^1 - \Lambda^0\right\| > \epsilon$, continue iterations returning to step 5(b).

Note that the integral in step (9) needs to be evaluated numerically. If the exogenous state variable follows a discrete process, the integration is done using the Markov transition matrix. If the exogenous state variable is a continuous process usual tools of numerical integration apply, typically this will involve discretizing the continuous process using, e.g., the Tauchen, Rouwenhorst, or Farmer-Toda methods. For normal shocks, quadrature methods tailored for purpose, such as Gauss-Hermite quadrature, are particularly powerful and lead to highly accurate results with minimal computational burden.

2.1.4 Extension to finite horizon

Binder et al. (2000) provide a comprehensive treatment for using Chebyshev parameterized expectations in a finite horizon context. The main difference from the infinite horizon is that the optimal conditional expectation function, $\Psi^*\left(u, z\right)$, is no longer time-invariant. As a result the approximation functions also depend on time. That is, at any time s of the life cycle

$$g^s\left(\cdot; \Lambda^s\right) \neq g^{s+1}\left(\cdot; \Lambda^{s+1}\right).$$

Step 6 of C-PEA then becomes:

$$\Upsilon\left[g^s\left(u, z; \Lambda^s\right), u', x, u, z\right] = 0. \tag{6}$$

And step 8 of C- PEA becomes:

$$\Upsilon\left[g^{s'}\left(u', z'; \Lambda^{s'}\right), u'', x', u', z'\right] = 0.$$

Like other OLG solution methods, we need to start from the last decision period by imposing a terminal condition for the endogenous state variables u and iterating backwards. The solution to the individual problem involves an array of coefficients that vary along the life cycle: $\Lambda^{*1}, \Lambda^{*2}, \ldots, \Lambda^{*S-1}$, where S is the number of generations.

Solutions to the problem depend on the life cycle, which means the individual problem needs to be solved repeatedly for each generation, unlike infinite horizon problems. As a result, the efficiency of the solution method becomes crucial as computational costs can

accumulate significantly. However, due to backward induction, the coefficients for the next period are given, leading to faster convergence compared to infinite horizon problems that require finding a fixed point.

One other difference that often separates OLG models from infinite horizon problems is the presence of a retirement period, during which a fixed income is available to agents. They also face a terminal decision period where they consume all the remaining capital and next period's assets are zero. This non-linearity creates a challenge for Chebyshev polynomials, but can be handled easily by ramping up the polynomial degree.[9] For models without idiosyncratic risk during retirement periods, increasing the polynomial degree is virtually costless in terms of computation.

We make an additional adjustment to the Chebyshev parameterized expectations algorithm when applying it to finite horizons. This modification involves utilizing an expanded Chebyshev grid. By employing this grid, we are able to include the value of zero in our interpolation grid. The points z_i on the expanded Chebyshev grid can be found using the formula below:[10]

$$z_i = \sec\left(\frac{\pi}{2m}\right)\cos\left(\frac{2i-1}{2}\frac{\pi}{m}\right), \qquad i = 1, \ldots, m.$$

2.1.5 Inequality constraints

Overlapping generations models typically involve a borrowing constraint where assets cannot fall below a certain threshold. Further, certain models require checking if control variables become negative when they should not. Inequality constraints do not pose an additional challenge under our method, once the Kuhn-Tucker conditions are written to involve relevant Lagrange multipliers associated with the inequality constraints within (3). One can then approach the problem in a typical way. First, the problem can be solved by imposing that the inequality constraints are not binding. If the solution implies that the constraints are binding, then the decision variables can be revised accordingly and the values of slack variables are implied by the system of equations (3).[11]

2.1.6 Policy and value functions under C-PEA

C-PEA is far from the only method of solving life-cycle models. Most other methods involve solving for the value function and/or the policy function and returning this as the 'solution'. While C-PEA returns the conditional expectation as the solution, the policy functions at any given point in the state space) can easily be found from the same first-order conditions in (3) using the approximated conditional expectation.

Certain problems, such as welfare comparisons, require value functions to be computed. Once the solution is found by C-PEA, the value functions can be computed in one step. Suppose, for generation s, the C-PEA solution yields $\Psi_s^*(u,z) = g_s(\Lambda_s^*, u, z)$ for the solution to the expectational function, where, as before, u and z are endogenous and exogenous state variables. The value function rule for generation s can be obtained via the steps below:

---

9 See Figure 8 in the appendix.

10 See Judd (1998, p. 222).

11 An easier, and computationally less demanding way to deal with inequality constraints is to replace the Kuhn-Tucker conditions with what is known as the Fisher-Burmeister equation. See Maliar et al. (2021) for details.

1. Choose polynomial degrees n_V and grid size m_V, over which the Value function is approximated.[12]

2. Create interpolation nodes r for the state variables as zeros of Chebyshev polynomials.

3. Transform r to the Chebyshev domain of $[-1, 1]$ using the same bounds $[a, b]$ used for approximating the expectational function.

$$\bar{r} = \frac{r+1}{2}(b-a) + a.$$

4. Create an $m_v \times n_v$ regression matrix Ψ. The rows are Chebyshev polynomials of degree 1 to n evaluated at r.

5. For $(u, z) \in \bar{r}$, use the optimal rule $g_s(u, z; \Lambda_s^*)$ in (6) to obtain the values for optimal control variables, x^*.

6. Calculate the value, V^s, given next period's value V^{s+1}:[13]

$$V^s(u_i, z_i) = U(x^*) + \beta V^{s+1}\left(u', z'\right).$$

7. Obtain coefficients Λ_V^{s*} by regressing V^s on Ψ.

The procedure above does not involve any iteration. As a result, the computational costs are negligible. Once, Λ_V^{s*} are obtained, we can evaluate the value function anywhere in the state space.

2.2 Computing the distribution of agents

There are then two standard approaches we could use to compute the distribution of agents: (i) Monte Carlo simulations, or (ii) iteration on the agent distribution. We choose to follow the iteration approach.[14]

We discretize the agent distribution on a grid. Because the optimal expectational function does not take values on this grid, we interpolate it onto the grid points below and above the value of the policy function, with linear weights used to assign probabilities.[15] In this way the policy function and the transition matrix on exogenous shocks can be combined to generate a transition matrix on the agent distribution (for each age, as the policy differs by age). Starting from an initial agent distribution at age 1, μ_1, we can then simply multiply by the transition matrix on the agent distribution iteratively to get the full agent distribution. This method of linear interpolation to create a transition matrix on a discretized agent distribution is entirely standard.

[12] These can be different from the polynomial degree and the grid size used for approximating the expectational function.

[13] The value for the next period, V^{s+1}, has already been computed, as a result of backward induction.

[14] For a one endogenous state model, as here, iteration tends to be faster and more accurate. Iteration is also more memory intensive and this becomes problematic in higher-dimensional models.

[15] If x is the value of the policy function and x_0 the grid point below, and x_1 the grid point above, then the probability of the lower grid point is $(x - x_0)/(x_1 - x_0)$ and the probability of the upper grid point is $(x_1 - x)/(x_1 - x_0)$.

2.3 Model aggregates and general equilibrium

Evaluating model aggregates, such as aggregate capital, is straightforward because the agent distribution has been discretised. We simply evaluate the approximated expectational functions on this grid and calculate model's endogenous variables, and then take a weighted sum, where the weights are from the agent distribution.

Since the agent distribution has been discretized evaluating model aggregates, such as aggregate capital, is a simple matter of evaluating a function on this grid, and then taking a weighted sum (with weights coming from the agent distribution).

We compute general equilibrium using a shooting-algorithm by updating the prices and endogenous parameters using the general equilibrium conditions. This is a standard and widely used approach. Computing model aggregates and finding general equilibrium when using C-PEA is the same as with many other standard algorithms for solving S-OLG models.

3. Example application

To demonstrate the use of C-PEA in solving S-OLG models we solve the model from page 536 of Heer and Maussner (2009). A period in the model corresponds to a year. The model features overlapping generations of households that perform consumption-savings and consumption-leisure decisions in the face of idiosyncratic productivity shocks. The production side of the economy consists of perfectly competitive markets and a representative firm with a Cobb-Douglas production function. The government uses taxes to fund pensions and general government spending and maintains a balanced budget.

Generations: There are $S = 70$ generations at a given time t.

Retirement: Households work for the first $T^w = 45$ periods of their lifetime, and retire at the age of $T^w + 1$. When they retire, they earn a pension $\bar{b}$.

Efficiency adjusted wages: During their working years, households earn the basic wage rate, w_t^0, adjusted by their efficiency level:

$$w_t = w_t^0 \bar{y}^s \epsilon_i z_t \qquad i = 1, \ldots, \theta,\ s = 1, \ldots, T^w.$$

The worker has a permanent efficiency type, ϵ_i, to accommodate features of ex-ante heterogeneity. There is also a idiosyncratic productivity shock, z_t, that follows a Markov process. Additionally, wages have an age-based profile, $\bar{y}^s$.

Government: The government collects taxes at a rate τ_w on wage income to finance total pension payments.

Pensions: The pension is equal to a fraction (replacement rate) ζ of the average wage income at the non-stochastic steady state: $\bar{b} = \zeta w \bar{n}$.

Households' problem: Households enjoy consumption, c_t^s, and leisure, l_t^s. They save by accumulating capital, k_t^s. They maximise lifetime utility when they are

at age 1 with respect to their lifetime budget constraints.

$$\begin{aligned}
\max_{\{c_s^t, l_s^t, k_{s+1}^{t+1}\}_{s=1}^S} \quad & \mathbb{E}_t \sum_{s=1}^{S} \beta^{s-1} \frac{\left(\left(c_{t+s-1}^s\right)^\gamma \left(l_{t+s-1}^s\right)^{1-\gamma} \right)^{1-\eta} - 1}{1-\eta} \\
\text{s.t.} \quad & k_{t+1}^{s+1} + c_t^s \leq (1+r_t)k_t^s + (1-\tau_t^w) w_t \epsilon_i z_{j,t} \bar{y}^s n_t^s, \quad s = 1, \ldots, T^w. \\
& k_{t+1}^{s+1} + c_t^s \leq (1+r_t)k_t^s + \bar{b}_t, \quad s = T^w+1, \ldots, S. \\
& k_{s+1}^{t+1} \geq 0, \quad s = 1, ..., S \\
& n_s^t + l_s^t = 1, \quad s = 1, ..., S
\end{aligned} \tag{7}$$

The total time endowment is 1, and is either spent on leisure, or employment, n. After retirement, $n = 0$, and a pension payment of $\bar{b}$ is received. Note that the leisure decision is trivial in retirement.

Production: There are a continuum of competitive firms with constant-returns-to-scale Cobb-Douglas production functions. Hence output in this economy is given by a representative firm with a Cobb-Douglas production function,

$$Y_t = A_t K_t^\alpha N_t^{1-\alpha},$$

where Y_t is output, A_t is technology, K_t is capital and N_t is effective labor supply. In general equilibrium the factor prices follow from the representative firm's profit maximisation:

$$\begin{aligned}
w_t &= (1-\alpha) A_t N_t^{-\alpha} K_t^\alpha, \\
r_t &= \alpha A_t N_t^{1-\alpha} K_t^{\alpha-1} - \delta,
\end{aligned} \tag{8}$$

where, δ denotes the capital depreciation rate, w_t is wage, and r_t is the interest rate.

Aggregation: Aggregate variables are given by the following:[16]

$$\begin{aligned}
K_t &= \sum_{s=1}^{S} \sum_{i=1}^{\theta} \int_k \int_z f(s, \epsilon_i, k, z)\, dz\, dk, \\
C_t &= \sum_{s=1}^{S} \sum_{i=1}^{\theta} \int_k \int_z c(s, \epsilon_i, k, z)\, f(s, \epsilon_i, k, z)\, dz\, dk, \\
N_t &= \sum_{s=1}^{T^w} \sum_{i=1}^{\theta} \int_k \int_z n(s, \epsilon_i, k, z)\, \epsilon_i z \bar{y}^s f(s, \epsilon_i, k, z)\, dz\, dk. \\
B_t &= \bar{b}_t \frac{S - T^w}{S},
\end{aligned} \tag{9}$$

where C_t is consumption, and B_t is total pension spending. Notice that N_t is the supply of effective labor units, not the aggregate fraction of time worked.

[16] We will later set z, the markov shock, to be finite-valued. So the integrals over z become sums.

3.1 Stationary equilibrium

At the stationary equilibrium, there is no aggregate uncertainty, $A_t = 1$, and the following conditions hold:

1. Households' solve the household problem given in (7).
2. The distribution function $F(s, \epsilon, k, z)$ associated with densities $f(s, \epsilon, k, z)$ are constant and consistent with the dynamics:

$$F(s+1, \epsilon_i, k', z') = \int_z \int_{k'=k'(s,\epsilon_i,k,z)} g(z'|z) F(s, \epsilon_i, k, z)\, dk\, dz.$$

3. Aggregate variables are given by conditions (9) and the below aggregate resource constraint holds:

$$C + \delta K = Y.$$

4. Factor prices are given by equations (8).
5. The government budget constraint is balanced:

$$B = \tau^w\, w\, N.$$

3.2 Calibration

The model is calibrated according to Heer and Maussner (2009). The idiosyncratic shock is assumed to follow a two-state Markov process, such that $z_{j,t} = [0.727, 1.273]$ with transition matrix:

$$p = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}.$$

The age-specific component of worker efficiency, $\bar{y^s}$, is taken from Hansen (1995), interpolated across generations, and centered around 1.

The other parameters are calibrated as in Table 1.

Table 1 – Calibration

Parameter	Explanation	Value
β	Discount factor	0.99
γ	Weight of consumption	0.28
η	Risk aversion	2
δ	Depreciation rate	0.08
α	Weight of capital	0.35
ζ	Replacement rate	0.30
ϵ	Efficiency types	[0.57 1.43]
T^w	Working years	45
S	Lifespan	70

We also assume that there are no bequests so that $K_{71} = 0$, and the agents start their life with no assets. Further, an equal measure of agents are born each period with efficiency $\epsilon_i z_j$. Therefore, the distribution for an age-1 household with type $\epsilon_i z_j$ has a mass $\frac{1}{4\times 70}$ at $k = 0$ and $z = z_j$.

3.3 Solution

We implement two C-PEA solutions for the model of the previous section. The solutions differ in terms of their solution parameters as summarised in Table 2. Our aim is to show that C-PEA can achieve a good mix of speed and accuracy by choosing the solution parameters according to the problem at hand.

Table 2 — Solution parameters, fast and slow solutions

	Solution 1	Solution 2
Polynomial degree, working agents	7	70
Polynomial degree, retired agents	70	70
Distribution grid size	25	200
Homotopy parameter, individual problem	0.9	0.9

The solutions differ in terms of polynomial degree and the size of the distribution grid. In the first and second solutions, the polynomial degree is set to 7 and 70, respectively, for all generations. Another distinction between the two solutions lies in the scale of the distribution grid. In the second solution, the distribution grid is set at 200 points, which is about three times the number of generations in the model. This follows the conventional approach in the literature on modelling heterogeneous agents. However, in solution 2, the distribution grid is kept at a smaller size of 25 points. We later demonstrate that this coarser grid can yield comparable accuracy under C-PEA if the grid points are selected thoughtfully. Using this coarser grid speeds up the computation process.

For the current example, we divide the asset space into two sub-spaces, $[0, 0.5]$ and $[0.505, 9]$ for the first permanent type, and $[0, 1]$ and $[1.055, 16]$ for the second permanent type. The first interval, in which the borrowing constraint binds more frequently compared to the second interval, comprises 10 of the 25 grid points for each permanent type.

3.3.1 Run times

Table 3 reports the run times associated with these solutions.[17,18] The first solution delivered a general equilibrium solution in under 3 seconds. The second solution provided a more accurate solution still with considerable speed.

Table 3 — Run times in seconds

	Solution 1	Solution 2
Life cycle permanent type 1	0.05	0.36
Life cycle permanent type 2	0.03	0.29
Distribution one step	0.08	2.49
General equilibrium	2.78	13.48

Note: Refer to Table 2 for solution parameters.

[17] Computer specifications: Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz 2.21 GHz, 16GB of RAM.

[18] In these examples, We started directly with the highest desired polynomial degree. For models where the individual problem is more complicated, it might be necessary to start with a low degree and increase the polynomial degree to the maximum desirable degree gradually by using coefficients from lower degrees as starting values.

The run-time results suggest that the C-PEA offers a plausible alternative to existing methods. Also, C-PEA offers the possibility of solving higher dimensional problems where other approaches might break down.

In the preceding examples, we simplified the first-order conditions to expedite computations, circumventing the need for a nonlinear equation solver. While this approach generally holds true across various solution methods, its impact is particularly pronounced in our specific method. Our iterative process involves solving the system of first-order conditions numerous times- both in the life cycle problem and during distribution steps. Employing a nonlinear equation solver for these solutions can significantly increase time costs. In our experiments, using a nonlinear solver would have increased the time required to find the general equilibrium by over 25-fold, averaging over 80 seconds per solution under Solution 1.

In more complex models, we advocate for using an economized distribution grid similar to Solution 1. Another effective strategy involves employing varying polynomial degrees, with a focus on problematic regions in the life cycle problem where borrowing constraints frequently bind. While we did not implement starting from good initial guesses for general equilibrium parameters in our examples, doing so in more demanding models could substantially reduce runtime.

An additional option to mitigate time costs is parallel computing. However, given that the solution to life cycle allocations relies on time iterations, any parallelization must occur within age groups. One potential approach is to parallelize across the Chebyshev grid, though we found that this method is often burdened by overhead costs, resulting in comparable runtimes. This observation may not hold true for more intricate models with multiple state variables. Alternatively, parallelizing the solution process across permanent types- particularly in models involving ex-ante heterogeneity like ours can yield almost linear reductions in runtime across permanent types, as expected.

As shown in Table 2, solutions 1 uses a compact yet uneven distribution grid, whereas solution 2 employs a standard grid size. While the effectiveness of the small distribution grid size is evident from Table 3 and Figure 3 as it maintains high accuracy with substantial speed gains, its benefits are likely to be more significant in more complicated models.

3.4 Comparison of aggregate and intergenerational variables in the stationary equilibrium

The values of aggregate variables are presented in Table 4 for the two solutions. The fast solution slightly overestimates aggregate capital stock and the real wage, albeit by a small margin. All other variables remain indistinguishable up to the fourth digit. Variables along the life cycle paint a similar picture. As seen in Figure 2, the two solutions are difficult to differentiate with the naked eye.

We conclude that the solution with a seventh degree polynomial degree and a compact distribution grid of size 25 performs comparably to the solution with a seventieth degree polynomial and a larger distribution grid, while reducing computation time to achieve general equilibrium by a factor of 5.[19]

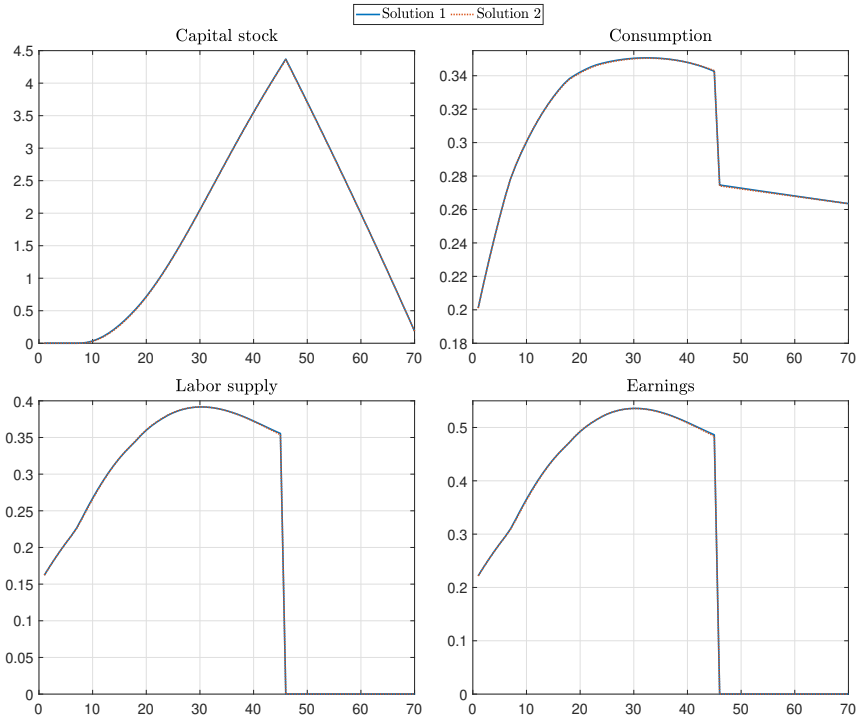
[19] An alternative approach could be to utilise an element grid, following the examples outlined in the appendix.

Table 4 — Aggregate variables in general equilibrium

	Solution 1	Solution 2
C	0.3039	0.3038
K	1.7765	1.7728
N	0.2119	0.2118
n	0.2041	0.2040
Y	0.4460	0.4456
r	0.0079	0.0080
w	1.3680	1.3673
τ^w	0.0935	0.0935
Earnings	0.2899	0.2896
$\bar{b}$	0.0759	0.0758

Note: Refer to Table 2 for solution parameters.

Figure 2 — Key variables across the life cycle



Note: Refer to Table 2 for solution parameters.

3.5 Accuracy according to Euler equation errors

We compare the accuracy of the solutions by calculating the associated Euler equation errors in the spirit of Christiano and Fisher (2000). The errors are calculated on a simulated path as well as on a circle similar to Juillard and Villemot (2011).

3.5.1 Accuracy on a simulated path

Definition: *Euler equation errors*

1. Collect Chebyshev polynomial parameters Λ_s^* from the C-PEA solution to

the lifecycle block of the stochastic OLG model.

$$\Upsilon\left[\mathbb{E}\left[\psi\left(x_{s+1}, u_{s+1}, z_{s+1}\right)|u_s, z_s\right], u_{s+1}, x_s, u_s, z_s\right] = 0$$

2. Draw a series of idiosyncratic shocks z of size $S \times 1$.
3. Starting from $s = 1$, compute x_s^* and u_{s+1}^* from model's system of equations:

$$\Upsilon\left[g\left(u_s, z_s; \Lambda_s^*\right), u_{s+1}^*, x_s^*, u_s, z_s\right] = 0.$$

4. Calculate the implied value of the next period's expectational function under Λ_s^*, denoted $\Phi\left(u_s^*, z_s\right)$. Note that

$$\Phi\left(u_s^*, z_s\right) = \int_{-\infty}^{\infty} \psi\left(x_{s+1}^*, u_{s+1}^*, z_{s+1}\right) f\left(z_{s+1} \mid z_s\right) dz,$$

where x_{s+1}^* solves

$$\Upsilon\left[g\left(u_{s+1}^*, z_{s+1}; \Lambda_{s+1}^*\right), u_{s+2}^*, x_{s+1}^*, u_{s+1}^*, z_{s+1}\right] = 0.$$

5. Define the residual $\mathcal{R}$ as:

$$\Upsilon\left[\Phi\left(u_s^*, z_s\right), u_{s+1}^*, x_s^*, u_s, z_s\right] + \mathcal{R} = 0.$$

6. Define the *Euler equation error*, $\mathcal{E}$, as the consumption compensation that sets $\mathcal{R} = 0$.

In practice, we initially construct a 70×1000 matrix of shocks. Subsequently, for each column of this matrix, we simulate the model forward from ages 1 to 70 using the approximated expectational function. At each age, we determine the next period's expectational function for each possible realisation of the shock, using the approximated function for the subsequent period. Finally, we compute the compensating variation in consumption, denoted as $\mathcal{E}$, necessary to equate the Euler equation to zero. In our simplified model, we arrive at the following expression for $\mathcal{E}_s$:

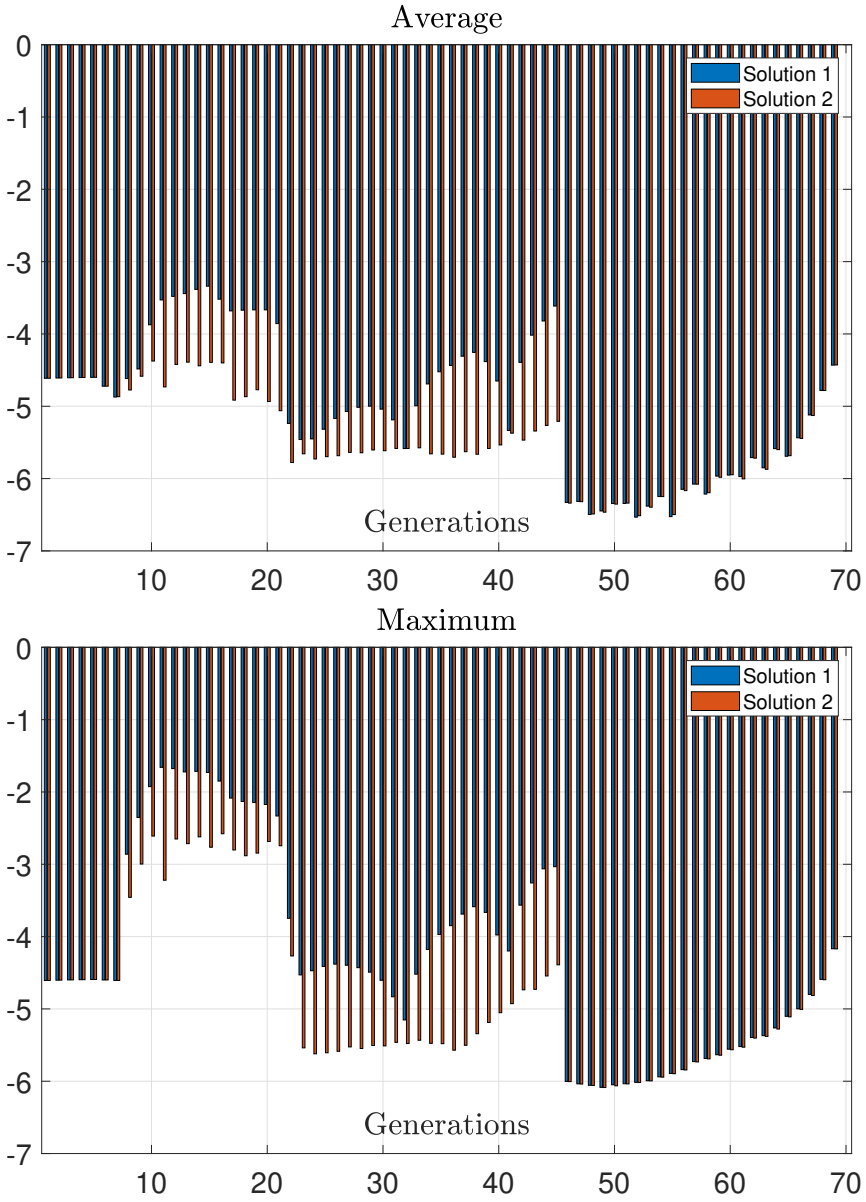
$$\mathcal{E}_s = \frac{\left[\frac{\beta\Phi_s^* + \mu_s^*}{\gamma(1-n_s^*)^{(1-\gamma)(1-\eta)}}\right]^{\frac{1}{\gamma(1-\eta)-1}}}{c_s^*} - 1. \tag{10}$$

Above, μ_s^* is the slack variable for the borrowing constraint. For the model at hand, $\Phi^*(z, k^{*,s}) = \sum_{j=1}^{2} p_j(1+r)\Omega^{s+1}\left(z_j', k^{*,s+1}\right)$, where Ω denotes the Lagrange multiplier on the budget constraint.

Our definition of Euler equation errors differs slightly from that in Christiano and Fisher (2000). In their framework, the non-negativity constraint is applied to investment, making it sensible to compute Euler equation errors while keeping Tobin's q constant. However, this approach would be arbitrary in our case. We have verified that our Euler equation errors differ only negligibly from those calculated according to their definition.

The Euler equation errors are reported in Figure 3. Both solutions yield reasonably accurate results. The average errors obtained by the faster Solution 1 are comparable to Solution 2, which employs a greater polynomial degree.

Figure 3 – Euler equation errors along a simulated path $\left(\log_{10}|\mathcal{E}|\right)$



Note: Refer to Table 2 for the solution parameters.

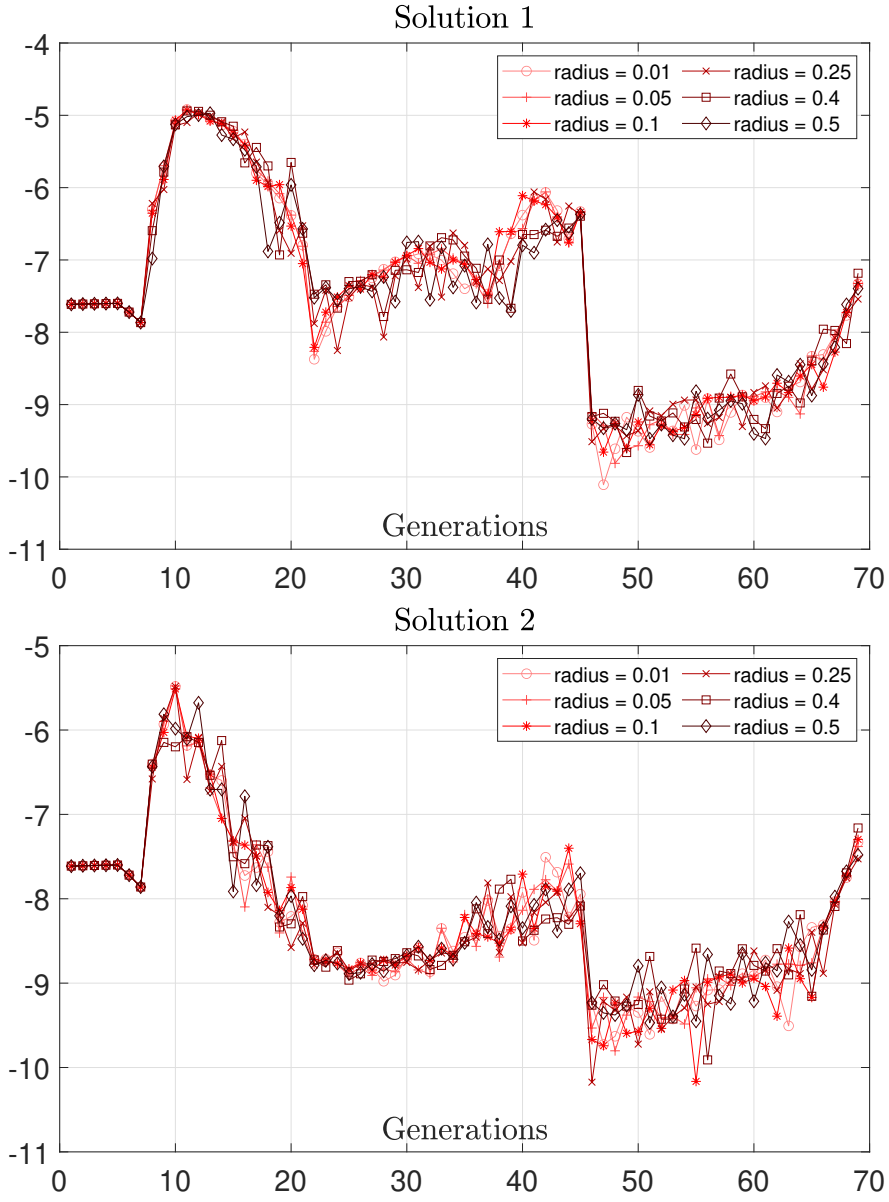
We can observe from Figure 3 that our solutions exhibit visibly higher maximum errors occurring between generations 8-21. This is attributable to two factors. First, the borrowing constraint binds more frequently at these ages because agents accumulate wealth at an aggressive rate. This non-linearity impacts the Chebyshev-PEA. The second reason is associated with the distribution of grid points. The distinctive life cycle nature of overlapping generation models implies that the capital stock spans different ranges at different ages. Using a single wide grid for all ages means that certain grid points are located in regions where agents never choose to allocate capital.[20] We hypothesize that using a Chebyshev grid with lower and upper limits varying across ages could further enhance solution accuracy by more efficiently allocating grid points. However, this would increase code complexity, particularly in computing distributions. We defer this task to future research.

[20] We achieved improved solution accuracy by employing an expanded Chebyshev grid that includes a capital stock of zero as the first element of the grid.

3.5.2 Accuracy on a circle

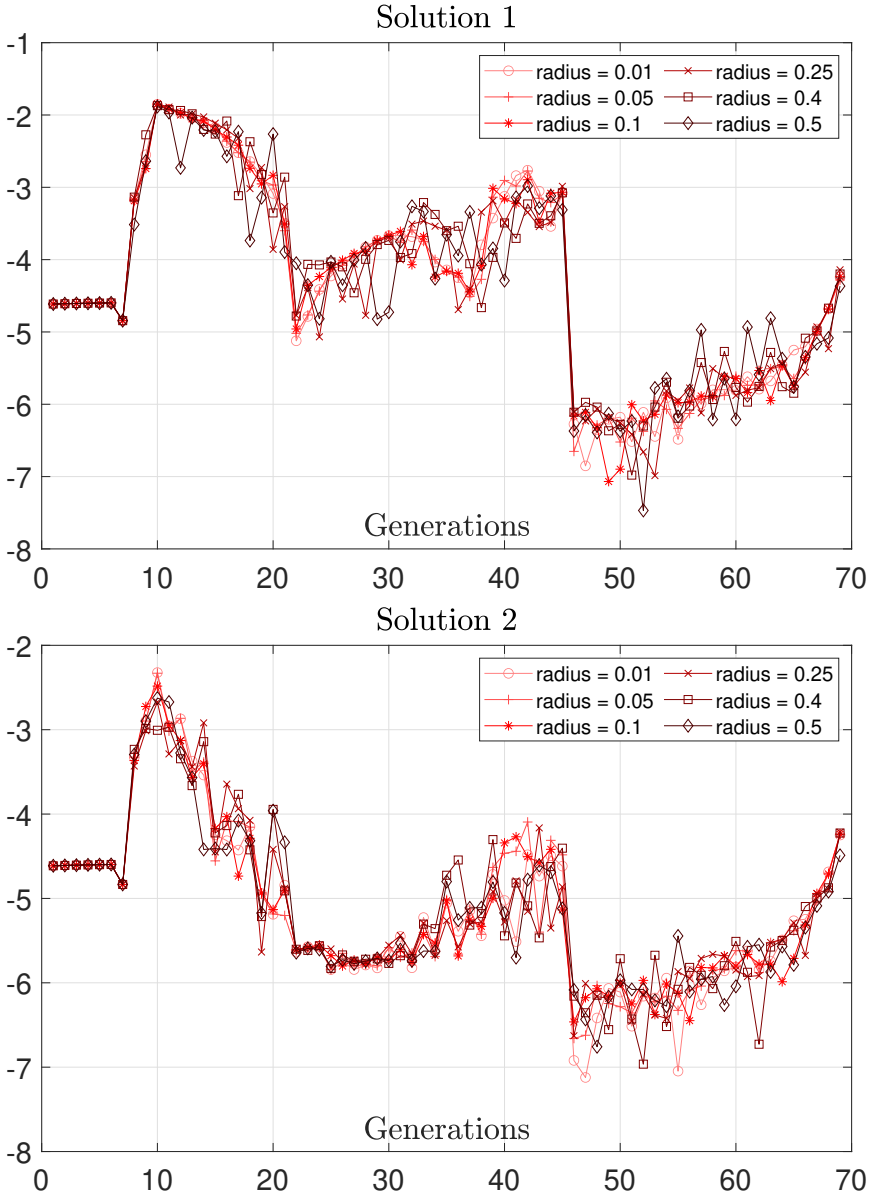
Our second measure of accuracy evaluates the performance of solutions when the model deviates from the steady state. To achieve this, we generate 1000 random asset values for each age, where these asset values deviate from the steady state on average by $radius \times 100$ percent. Figure 4 presents the average errors, defined in equation (10), for $radius \in 0.01, 0.05, 0.10, 0.25, 0.40, 0.50.$.

Figure 4 — Average Euler equation errors on a circle $(\log_{10}|\mathcal{E}|)$



Note: Refer to Table 2 for the solution parameters.

Figure 5 — Maximum Euler equation errors on a circle $\left(\log_{10}|\mathcal{E}|\right)$



Note: Refer to Table 2 for the solution parameters.

Figure 4 illustrates the global nature of the solutions. Since we solve the model on a sufficiently wide grid where capital never exceeds the upper bound at any age, the solution does not significantly deteriorate as we move away from the steady state. The maximum errors depicted in Figure 5 confirm the findings from Figure 3 that the solutions exhibit higher maximum errors for ages 8-21.

3.6 Comparison to pure discretization solution

Next, we re-solve the model using pure discretization methods (implemented in VFI Toolkit).[21] We report the C-PEA solution again in the first column of Table 5. The second and third columns report the difference between the C-PEA solution and the pure discretization solution measured as the percent difference and as the absolute difference.[22] For the life-cycle profiles, which are a number for each age, we report the maximum across

[21] See https://www.vfitoolkit.com.

[22] The percentage difference is reported as a percentage of the pure discretization solution.

Table 5 — Table of model values and differences from pure discretization solution

	Model	% Difference	Absolute difference
General equilibrium			
Interest rate (r)	0.0080	-13.21	-0.0012
Wage (w)	1.3672	0.74	0.0100
Tax rate (τ_w)	0.0935	-12.07	-0.0128
Wage ($\bar{b}$)	0.0758	-11.38	-0.0097
Aggregates			
Consumption (C)	0.3037	0.79	0.0024
Capital Stock (K)	1.7728	2.89	0.0498
Effective Labor Supply (N)	0.2118	0.76	0.0016
Output (Y)	0.2040	1.50	0.0066
Life-Cycle Profiles (max difference over ages)			
Consumption	-	2.53	0.0083
Assets	-	5.00	0.2078
Earnings	-	3.75	0.0177
Lorenz curves (max difference over percentiles)			
Earnings	-	0.50	0.0012
Assets	-	7.97	0.0068

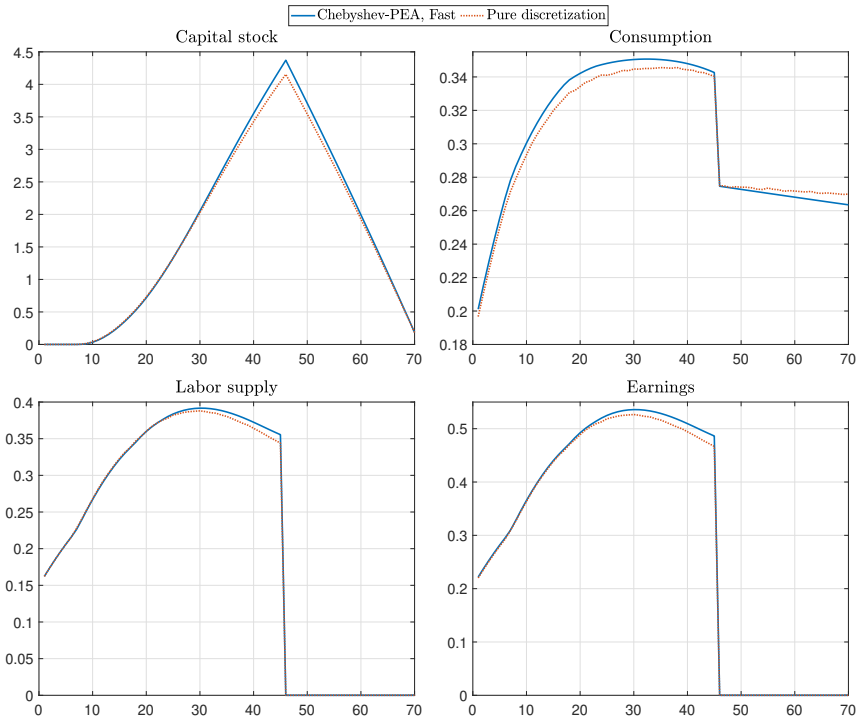
Notes: The second and third columns report on accuracy, measured as the percentage absolute difference and absolute difference between our solution and a pure discretization solution of the same model using 101 points for labor and 1301 points for assets.

the ages of each of these two difference measures. In general, the different solution methods yield very similar values, particularly in absolute terms, and the solutions have very similar life-cycle properties. The life cycle profiles of key variables are depicted in Figure 6.

We also computed Lorenz curves and report the maximum across the percentiles of each of these two difference measures.[23] As can be seen the differences between the two solutions are small enough to be of no concern; while 3% may sound non-trivial it is very demanding in this context and smaller than the most accurate of the fourteen papers replicated in Kirkby (2022), so seems sufficiently accurate in practice.

[23] Pure discretization has known convergence properties for this class of models and so, as long as a sufficiently large grid is used, is known to give the correct solution. See Kirkby (2022) for detailed discussion of this issue.

Figure 6 – Comparison to pure discretization



Note: "Chebyshev-PEA, Fast" refers to Solution 1 in Table 2.

4. Conclusion

While S-OLGs are arguably the most appropriate tools to investigate distributional and intergenerational consequences of fiscal policies, solving these models for policy analysis remains a highly challenging task. The lifecycle block of these models must be solved numerous times to reach a general equilibrium solution[24] under significant nonlinearities due to the finite lifetime nature of the model and the presence of borrowing constraints. In practice, this necessitates constraining the dimensionality of the model, such as limiting the number of generations, facets of ex-ante heterogeneity, and endogenous states. Consequently, policy analysts are often limited to using simpler S-OLG models with fewer features and less realism.

In this paper, we demonstrate that employing C-PEA for solving S-OLGs can be a highly efficient approach. We provide a detailed explanation of the method and illustrate its practical application in solving a prototypical S-OLG model.

Our model application demonstrates that the computational requirements are low and the run times are fast using C-PEA. While Chebyshev polynomials are more suitable for approximating smooth functions, the method can handle nonlinearities in S-OLGs reasonably well by increasing the polynomial degree. This approach is feasible in a finite lifetime setting because the decision rule is not time-invariant, eliminating the need to find a consistent decision rule that remains unchanged between two time periods. Instead, we leverage backward induction, making it efficient to find decision rules via Chebyshev regressions since the next period's decision rule is simply taken as given.

The real strength of using Chebyshev-PEA to solve S-OLGs lies in the ability of Chebyshev

[24] Or even greater number of time to compute transitional dynamics.

polynomials to scale to higher dimensional problems (Judd, Maliar, Maliar and Valero 2014). This is a promising avenue considering the significant challenge of solving S-OLGs with multiple endogenous state variables. How well this can be exploited by C-PEA to solve multi-dimensional S-OLGs is a question for future research.

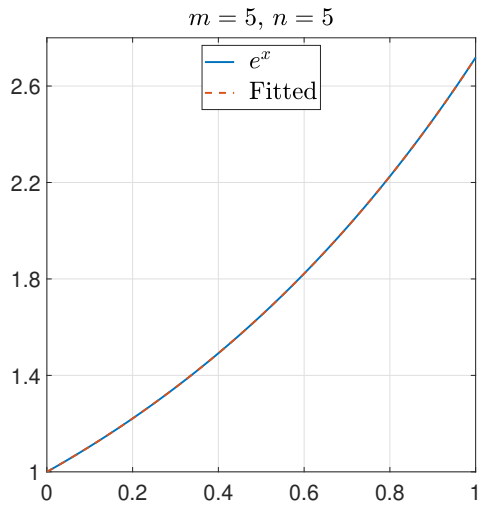
Appendix: Chebyshev regression algorithm in practice

The discussion above implies that Chebyshev polynomials are particularly suitable for approximating smooth functions. They can also do a reasonably good job for approximating functions with kinks, as reported in Judd (1998).[25] This is particularly important in the context of OLG models, because such models can display significant nonlinearities at the beginnings and ends of life cycles, and around the date of retirement.

In this section, we demonstrate the Chebyshev regression algorithm for approximation two functions, $f(x) = e^x$, and $f(x) = |x|$. The former is a smooth function, and the latter is a function with a kink. Admittedly, this exercise is much simpler than solving the dynamic and stochastic problems that we face in economics. Nevertheless, it will help us demonstrate a few important points about the Chebyshev regression algorithm, which is at the core of our solution method.

We approximate $f(x) = \exp(x)$ in Figure 7. As suggested by the discussion so far, Chebyshev polynomials approximate this smooth function with great success with just 5 interpolation nodes, and using Chebyshev polynomials up to degree 5.

Figure 7 – Approximation of e^x with Chebyshev regression algorithm



Next we approximate the function $f(x) = |x|$. This function has a kink at $x = 0$. In the northwest panel of Figure 8, the function is approximated with Chebyshev polynomials of degree 5 using their zeros as the interpolation grid as we did in Figure 7. As expected, the results of this exercise are much less accurate. In the northeast panel, the size of the grid is increased, while the polynomial degree is held constant. As seen in the figure, this helps by improving the approximation along the edges but does not help around the kink. If anything, the maximum error becomes greater.

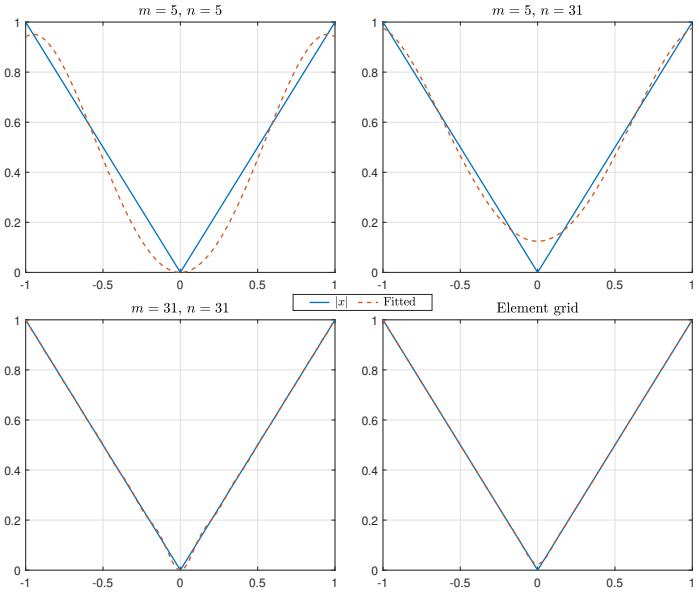
Next, we increase the polynomial degree along with the grid. As seen in the south-

[25] See Table 6.4 and Figure 6.9 therein.

west panel of Figure 8, Chebyshev polynomials are able to approximate this function successfully. The approximation is still very fast[26] and efficient.

For the problems we face in economics modelling, however, the time costs can increase drastically. First, it might not be possible to increase the polynomial degree to the levels observed in Figure 8 when the dimensionality of the problem increases. Second, depending on the complexity of the problem at hand, it might be necessary to start from a lower polynomial degree and increase it gradually using solutions from lower degrees as an initial guess. Third, in OLGs in particular, the solution needs to be repeated for each generation and as a result, computation costs may add up. In the southeast panel of Figure 8, we demonstrate applying an element grid to the Chebyshev regression. Here, we divide the solution space into three sub-intervals: $[-1, -0.2]$, $(-0.2, 0.2)$, and $[0.2, 1]$. In the middle, narrower sub-interval, which includes the kink, we choose $n = m = 8$. In the other sub-intervals, which span less-problematic regions, we choose $n = m = 3$. The approximation is quite good despite the fact that only 14 interpolation nodes are chosen and the polynomial degrees are not very high. Although we have not yet faced a problem that requires its application in our C-PEA algorithm, we have demonstrated a case with element grids to indicate that they are good alternatives when the problem displays a difficult kink.

Figure 8 – Approximation of $|x|$ with Chebyshev regression algorithm



[26] The approximation takes less than a hundredth of a second.

References

Binder, M., Pesaran, M.H., Samiei, S.H., 2000. Solution of nonlinear rational expectations models with applications to finite-horizon life-cycle models of consumption. Computational Economics 15, 25–57.

Binning, A., McKenzie, S., Özbilgin, M., Smith, C., 2024. Modelling the fiscal implications of New Zealand's ageing population using a large-scale overlapping generations model. New Zealand Treasury*Forthcoming*.

Cai, Y., Judd, K.L., Lontzek, T.S., Michelangeli, V., Su, C.L., 2017. A nonlinear programming method for dynamic programming. Macroeconomic Dynamics 21, 336–361.

Christiano, L.J., Fisher, J.D., 2000. Algorithms for solving dynamic models with occasionally binding constraints. Journal of Economic Dynamics and Control 24, 1179–1232.

den Haan, W.J., Marcet, A., 1990. Solving the stochastic growth model by parameterizing expectations. Journal of Business & Economic Statistics 8, 31–34.

Heer, B., Maussner, A., 2009. Dynamic General Equilibrium Modeling. Number 978-3-540-85685-6 in Springer Books, Springer. doi:10.1007/978-3-540-85685-6.

Judd, K., Maliar, L., Maliar, S., Valero, R., 2014. Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. Journal of Economic Dynamics and Control 44(C), 92–123.

Judd, K.L., 1998. Numerical Methods in Economics. volume 1 of *MIT Press Books*. The MIT Press.

Juillard, M., Villemot, S., 2011. Multi-country real business cycle models: Accuracy tests and test bench. Journal of Economic Dynamics and Control 35, 178–185. URL: https://ideas.repec.org/a/eee/dyncon/v35y2011i2p178-185.html.

Kirkby, R., 2022. Quantitative macroeconomics: Lessons learned from fourteen replications. Computational Economics doi:https://doi.org/10.1007/s10614-022-10234-w.

Krueger, D., Kubler, F., 2004. Computing equilibrium in olg models with stochastic production. Journal of Economic Dynamics and Control 28(7), 1411–1436.

Maliar, L., Maliar, S., Winant, P., 2021. Deep learning for solving dynamic economic models. Journal of Monetary Economics 122, 76–101.

Malin, B., Krueger, D., Kubler, F., 2011. Solving the multi-country real business cycle model using a smolyak-collocation method. Journal of Economic Dynamics and Control 35, 229–239.

Munos, R., Szepesvari, C., 2008. Finite-time bounds for fitted value iteration. Journal of Machine Learning Research 1, 815–857.