https://doi.org/10.26219/heal.aueb.9687

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΗΣ ΕΠΙΣΤΗΜΗΣ

Local vs Global Solution Methods in Real Business Cycle Models: A Comparative Analysis of Perturbation, Value Function Iteration, Endogenous Grids, and Neural Networks

ΙΑΣΟΝΑΣ ΓΚΑΪΓΚΕΡ

Διπλωματική εργασία υποβληθείσα προς μερική εκπλήρωση των απαραιτήτων προϋποθέσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης

Αθήνα [11,2025]

https://doi.org/10.26219/heal.aueb.9687

Επιβλέπων Καθηγητής: Αγγελής Αλεξόπουλος, ΟΠΑ

Εξεταστής καθηγητής:

Εξεταστής καθηγητής:

https://doi.org/10.26219/heal.aueb.9687

Table of Contents

https://doi.org/10.26219/heal.aueb.9687

https://doi.org/10.26219/heal.aueb.9687

https://doi.org/10.26219/heal.aueb.9687

Abstract

This thesis investigates and compares alternative numerical methods for solving the stochastic Real Business Cycle (RBC) model, focusing on the trade off between local analytical efficiency and global numerical accuracy. The analysis benchmarks four main approaches, Perturbation, Value Function Iteration (VFI), the Endogenous Grid Method (EGM), and Neural Network (NN)–based solvers, under a unified calibration based on Schmitt-Grohé and Uribe (2004).

The Perturbation method provides a fast and analytically transparent local approximation, yet its accuracy deteriorates under large shocks or high volatility. A systematic stress test quantifies the precise boundary of its validity, establishing thresholds for when global methods are required. The global solvers, VFI and EGM, compute policy functions directly on the full state space, offering benchmark accuracy; among them, EGM achieves near-perfect equilibrium precision with substantially reduced computational cost.

Building upon these, two Neural Network architectures are implemented: a hybrid NN(Global) trained via distillation from EGM followed by Euler-equation fine-tuning, and an autonomous NN(Equation) trained solely on the model's structural conditions. Both achieve global accuracy comparable to traditional methods, with NN(Equation) showing superior robustness in the tails of the state space. While Neural Networks entail significant upfront training time, they enable instantaneous policy evaluation after convergence, introducing a "solve-once, simulate-instantly" paradigm highly suitable for large-scale simulations and sensitivity analyses.

The results confirm that local methods remain sufficient near the steady state, EGM constitutes the optimal benchmark for most global analyses, and Neural Networks provide a credible and powerful alternative for high-dimensional or computation-intensive macroeconomic applications.

Η παρούσα διπλωματική εργασία διερευνά και συγκρίνει εναλλακτικές αριθμητικές μεθόδους επίλυσης του στοχαστικού μοντέλου Πραγματικού Επιχειρηματικού Κύκλου (RBC), δίνοντας έμφαση στον συμβιβασμό μεταξύ τοπικής αναλυτικής αποδοτικότητας και σφαιρικής αριθμητικής ακρίβειας. Η ανάλυση αξιολογεί τέσσερις κύριες προσεγγίσεις, την Perturbation, την Value Function Iteration (VFI), τη Endogenous Grid Method (EGM) και λύσεις βασισμένες

https://doi.org/10.26219/heal.aueb.9687

σε Νευρωνικά Δίκτυα (NN), κατά τη διάρκεια μιας ενοποιημένης βαθμονόμησης, σύμφωνα με το πλαίσιο που καθόρισαν οι Schmitt-Grohé και Uribe (2004).

Η μέθοδος Perturbation παρέχει μια ταχεία και αναλυτικά διαφανή τοπική προσέγγιση, η ακρίβεια της οποίας όμως μειώνεται υπό μεγάλες διαταραχές ή υψηλή μεταβλητότητα. Ένα συστηματικό stress test ποσοτικοποιεί τα ακριβή όρια εγκυρότητάς της, καθορίζοντας τα όρια πέρα από τα οποία απαιτούνται παγκόσμιες μέθοδοι. Οι παγκόσμιες λύσεις, VFI και EGM, υπολογίζουν τις συναρτήσεις πολιτικής απευθείας σε όλο τον χώρο καταστάσεων, προσφέροντας σημείο αναφοράς ως προς την ακρίβεια· μεταξύ αυτών, το EGM επιτυγχάνει σχεδόν άριστη ακρίβεια ισορροπίας με σημαντικά μικρότερο υπολογιστικό κόστος.

Βάσει αυτών, υλοποιούνται δύο αρχιτεκτονικές Νευρωνικών Δικτύων: ένα υβριδικό NN(Global), εκπαιδευμένο μέσω απόσταξης από το EGM και στη συνέχεια βελτιστοποιημένο με βάση την εξίσωση Euler, και ένα αυτόνομο NN(Equation), το οποίο εκπαιδεύεται αποκλειστικά στις δομικές συνθήκες του μοντέλου. Και τα δύο επιτυγχάνουν σφαιρική, παγκόσμια ακρίβεια συγκρίσιμη με τις παραδοσιακές μεθόδους, με το NN(Equation) να παρουσιάζει ανώτερη ανθεκτικότητα στα άκρα του χώρου καταστάσεων. Παρότι τα Νευρωνικά Δίκτυα απαιτούν σημαντικό χρόνο εκπαίδευσης, επιτρέπουν στιγμιαία αξιολόγηση των συναρτήσεων πολιτικής μετά τη σύγκλιση, εισάγοντας ένα παράδειγμα «λύσε μία φορά, προσομοίωσε άμεσα», ιδιαίτερα κατάλληλο για μεγάλης κλίμακας προσομοιώσεις και αναλύσεις ευαισθησίας.

Τα αποτελέσματα επιβεβαιώνουν ότι οι τοπικές μέθοδοι παραμένουν επαρκείς κοντά στη σταθερή κατάσταση ισορροπίας, το EGM συνιστά το βέλτιστο σημείο αναφοράς για τις περισσότερες σφαιρικές αναλύσεις, και τα Νευρωνικά Δίκτυα αποτελούν μια αξιόπιστη και ισχυρή εναλλακτική για μακροοικονομικές εφαρμογές υψηλής διάστασης ή αυξημένων υπολογιστικών απαιτήσεων.

Chapter 1. Introduction

Quantitative macroeconomics stands at the intersection of economic theory and computational practice. Its core objective is to translate abstract theoretical models of the macroeconomy, comprising the interactions of households, firms, and policymakers, into quantitative frameworks capable of replicating empirical data and informing policy decisions. This translation is fundamentally dependent on numerical methods that can approximate the equilibrium dynamics of nonlinear stochastic models, which rarely admit closed-form

https://doi.org/10.26219/heal.aueb.9687

solutions. The evolution of these methods, therefore, directly shapes the complexity and realism of the economic questions we can address.

Within this landscape, the Real Business Cycle (RBC) model has served for decades as the canonical laboratory for testing new computational techniques. Its clarity, analytical tractability, and well-understood dynamics provide an ideal benchmark. The methodological journey in solving such models has been one of navigating trade-offs. Early approaches prioritized analytical convenience and speed through local approximations, exemplified by Perturbation methods. These techniques linearise the model around a steady state, offering immense computational efficiency but at the cost of global accuracy. As the field's ambitions grew to encompass larger shocks, financial frictions, and heterogeneous agents, the limitations of local methods became a binding constraint, spurring the development of global solvers.

Global methods, such as Value Function Iteration (VFI) and the Endogenous Grid Method (EGM), address this limitation by computing policy functions across the entire state space. They offer benchmark accuracy but often at a significant computational cost, especially as model dimensionality increases. This trade-off between local efficiency and global robustness has defined the practitioner's choice for years. Recently, a new paradigm has emerged from the intersection of computational economics and machine learning: the use of Neural Networks as universal function approximators to solve economic models. These methods promise global accuracy without relying on predefined grids, potentially overcoming the curse of dimensionality that plagues traditional global methods. However, their performance, reliability, and computational profile relative to established techniques remain an active and critical area of investigation.

This thesis conducts a systematic, quantitative comparison of these alternative numerical approaches within a consistent stochastic RBC environment. We benchmark four classes of methods, local Perturbation, global Value Function Iteration (VFI), the highly efficient Endogenous Grid Method (EGM), and novel Neural Network (NN) solvers, under a unified calibration. Our goal is threefold: first, to precisely quantify the boundary where local Perturbation methods lose their validity under increasing volatility and shock sizes; second, to establish a modern hierarchy of global solvers based on both accuracy and computational cost; and third, to rigorously evaluate the promise of Neural Networks, assessing whether they represent a viable new tool for computational macroeconomics or merely a theoretical curiosity.

https://doi.org/10.26219/heal.aueb.9687

The analysis proceeds in a structured sequence. Chapters 2 and 3 lay the theoretical foundation, reviewing the relevant literature and formulating the one-sector RBC model used throughout the study. Chapter 4 forms the methodological core, detailing the implementation of each numerical solver. Chapter 5 introduces a novel stress-test framework that systematically measures the breakdown point of Perturbation methods, establishing quantitative reliability thresholds. Chapter 6 defines a unified set of performance metrics to ensure a fair and comprehensive comparison across all methods. Chapter 7 details the common calibration based on Schmitt-Grohé and Uribe (2004), while Chapter 8 presents the empirical results, evaluating all methods across a spectrum of volatility. Finally, Chapter 9 outlines promising directions for future research, and Chapter 10 concludes.

The contribution of this work is threefold. First, it moves beyond qualitative assumptions to provide a quantitative, empirically derived map of the domains of validity for local and global methods. Second, it establishes a clear and practical hierarchy among global solvers, confirming EGM's status as a benchmark for accuracy and efficiency. Third, it provides a rigorous, evidence-based assessment of Neural Network (NN) solvers, demonstrating their capacity for global accuracy and identifying their unique computational niche for large-scale simulation. By unifying traditional and modern approaches under a single experimental framework, this thesis aims to bridge the gap between analytical tractability and data-driven computational power, providing a clear guide for future applications in dynamic stochastic modeling.

Chapter 2. Literature review

Existing literature in computational macroeconomics has developed several numerical methods for solving dynamic stochastic general equilibrium (DSGE) and real business cycle (RBC) models. These approaches differ primarily in how they approximate the model’s equilibrium, balance accuracy and speed, and handle nonlinearities or constraints. The following review describes the main, most known techniques that have shaped this field, highlighting each method’s innovations and differences.

The Perturbation method (Judd, 1998; Schmitt-Grohé & Uribe, 2004) is a very precise and local approximation that linearises or expands the model’s equilibrium conditions around the deterministic steady state using Taylor series. It offers an analytical and computational efficient way to obtain local dynamics and depiction of the under-investigation economy. It is easy to

https://doi.org/10.26219/heal.aueb.9687

interpret, fact that explains its popularity in macroeconomic modelling. The methodological foundation in local approximations imposes a fundamental limitation, whereby its accuracy erodes markedly in response to large scale exogenous shocks or under strong nonlinear dynamics.

The Value Function Iteration (VFI) approach (Judd, 1998) directly solves the Bellman equation through dynamic programming. Unlike Perturbation method, Value Function Iteration (VFI) provides a globally accurate solution and can accommodate nonconvexities and occasionally binding constraints. Its main hindrance and significant challenge of Value Function Iteration (VFI) is the computational intensity that accompanies the method, which increases exponentially with the number of state variables.

The computational issue of Value Function Iteration (VFI) was mitigated by Carrol (2006), who developed the Endogenous Grid Method (EGM), an approach that subsequently reduces the computational expense of Value Function Iteration (VFI) by avoiding repeated computational costly maximisations. The method's efficiency stems from the use of a constructed endogenous grid of state variables, which achieves high accuracy and speed in models with continuous choices, making it especially suitable for consumption-savings, like the RBC this thesis focuses, and heterogeneous-agent frameworks.

Pushing the boundaries of computational economics in recent years, Neural Network (NN)–based methods have established themselves as a powerful class of global solution techniques. Fernández-Villaverde et al. (2020) proposed a continuous-time deep learning algorithm that embeds Neural Networks directly into the Hamilton–Jacobi–Bellman equation, training four interconnected networks—representing the value function, policy function, and Karush-Kuhn-Tucker multipliers—through mini-batch gradient descent. Their key innovation lies in eliminating the need for high-dimensional integration and achieving efficient global solutions to models with dozens of continuous state variables. Maliar, Maliar, and Winant (2021) introduced a complementary simulation-based approach that uses feed-forward Neural Networks to approximate policy functions globally. Their "all-in-one" algorithm minimizes Euler or Bellman residuals via stochastic gradient descent, focusing on the model's ergodic region rather than on predefined grids, which greatly enhances scalability. Finally, Kase (2022) developed a grid-free NN framework for DSGE models, in which policy functions are parameterized directly over the state and shock space. By minimizing Euler-equation residuals and enforcing equilibrium constraints through penalty terms, this method learns both optimal

https://doi.org/10.26219/heal.aueb.9687

policies and the ergodic distribution simultaneously. Collectively, these contributions mark a significant step toward integrating machine learning into macroeconomic modelling, offering flexible, data-driven tools capable of handling nonlinearities and high-dimensional state spaces beyond the reach of traditional methods.

Overall, the evolution from Perturbation to Value Function Iteration (VFI), Endogenous Grid Method (EGM), and finally Neural Network (NN)-based methods reflects a shift from local analytical approximations to global, data-driven approaches, expanding the frontier of what can be solved in modern macroeconomics by overcoming the curse of dimensionality, without risking accuracy and efficiency of the already known and widely used methods.

Chapter 3. The Real Business Cycle Model

In this chapter I give an overview of the model and its structure, the equilibrium conditions and way the Deterministic state is being calculated.

3.1 Real Business Cycle Framework

Real Business Cycle (RBC) models constitute a foundation of modern macroeconomic theory. These models describe the economy as driven by agents who exhibit rational behaviour and utility maximization. These agents respond to real (non-monetary) shocks—typically technological disturbances—that are being transmitted through capital accumulation, labour supply and productivity dynamics. RBC models capture how fluctuations in productivity, within a certain economy convert in output, consumption, investment, and employment movements over time.

The baseline RBC framework with a representative agent provides a simple yet powerful environment to observe and study the dynamics of aggregate variables, and how they react in different methods. The formation of the used RBC model is consistent with the set up presented by Schmitt-Grohé and Uribe (2004) and follows the standard neoclassical growth model.

3.2 Model Structure

The model consists of three main components: a representative household, a production technology, and a stochastic process for productivity. The household makes consumption and savings decisions to maximize its expected lifetime utility. The production function combines capital and productivity to produce a Cobb-Douglas technology output. The stochastic process

 https://doi.org/10.26219/heal.aueb.9687

introduces random shocks to productivity that generate the aggregate fluctuations in the economy. All agents behave competitively, and markets are complete.

In the numerical implementation that follows, the model is solved in logarithmic form for numerical stability and analytical convenience. Specifically, the variables are expressed as $\ln c$, $\ln k$, $\ln A$ rather than in levels. For simplicity, the baseline run in chapter 4 assumes full depreciation ($\delta = 1$) and independent and identically distributed (i.i.d.) technology shocks ($\rho = 0$). These assumptions imply a triangular state transition matrix in which productivity evolves exogenously and does not depend on capital accumulation. The general formulation of the model, however, allows for any $0 < \delta < 1$ and $0 < \rho < 1$ The latter model settings follow the Schmidt Grohe and Uribe (2004) exact implementation choices, to be able to compare the results across methods.

The representative household maximizes expected utility:

$$\max_{\{c_t, k'\}} \mathbb{E} \sum \beta u(c) , \qquad (3.1)$$

Where utility takes the constant relative risk aversion (CRRA) form:

$$u(c) = \frac{c^{1-\gamma} - 1}{1 - \gamma}, \quad (3.2)$$

With $0 < \beta < 1$ and $\gamma > 0$.

Technology and Resource Constraint

Output is produced using Cobb-Douglas technology:

$$y = Ak^{a}, \quad (3.3)$$

With $0 < \alpha < 1$

and the aggregate resource constraint is:

$$c + k' = y + (1 - \delta)k, \quad (3.4)$$

Or equivalently,

$$k' = (1 - \delta)k + Ak^{a} - c, \quad (3.5)$$

Where $\delta \in [0,1]$ denotes the depreciation rate of capital.

Stochastic Process for Technology

https://doi.org/10.26219/heal.aueb.9687

Productivity evolves according to an autoregressive AR(1) process in logs:

$$\log A' = \rho \log A + \sigma \varepsilon', \quad (3.6)$$

Where $\varepsilon' \sim N(0,1)$, $-1 < \rho < 1$ measures persistence, and $\sigma > 0$ controls volatility

3.3 Equilibrium Conditions

The household's first-order condition gives the Euler equation:

$$u'(c) = \beta \mathbb{E}_t[u'(c')R'], \quad (3.7)$$

Where the gross return on capital is defined as:

$$R' = aA'{k'}^{a-1} + 1 - \delta, \quad (3.8)$$

Equations (3.4), (3.6) and (3.7) jointly characterize the competitive equilibrium of the economy.

3.4 Deterministic Steady State

In the non-stochastic steady state ($A = A' = 1$), all variables are constant:

$$k = k' = k^*, c = c^*, \text{and } y = y^*.$$

The steady state equilibrium satisfies:

$$1 = \beta[\alpha(k^*)^{a-1} + 1 - \delta], \quad (3.9)$$

$$y^* = (k^*)^a, \qquad (3.10)$$

$$c^* = y^* - \delta k^* \quad (3.11)$$

From equation (3.9), the steady state capital stock can be expressed as:

$$k^* = \left(\frac{\alpha\beta}{1 - \beta(1 - \delta)}\right)^{\frac{1}{1-\alpha}}, \quad (3.12)$$

The equilibrium of the model is defined by a system of equations (3.4),(3.6) and (3.7), which combined generate the consumption, capital and productivity paths over time.

Using their functional forms, the equilibrium conditions can be restated as:

$$c^{-\gamma} = \beta \mathbb{E}\left[{c'}^{-\gamma}\left(aA'{k'}^{a-1} + 1 - \delta\right)\right], \qquad (3.13)$$

https://doi.org/10.26219/heal.aueb.9687

$$c + k' = Ak^a + (1-\delta)k, \quad (3.14)$$

$$\log A' = \rho \log A + \sigma\varepsilon', \qquad \varepsilon' \sim N(0,1). \quad (3.15)$$

The above system (3.13)-(3.15) fully characterize the dynamic stochastic equilibrium of the model for given initial conditions k and A.

The model's steady state when using the given parameter values from Schmitt-Grohé and Uribe (2004) goes as follows:

The baseline parameters are: $a = 0.3, \beta = 0.95, \gamma = 2, \delta = 1, \rho = 0$

At the deterministic steady state (A=1), the Euler condition reduces to

$$1 = \beta[\alpha k^{a-1}].$$

Solving for the steady state capital stock gives

$$k^* = (\alpha\beta)^{\frac{1}{1-\alpha}}$$

Given the baseline parameters,

$$k^* = 0.1664, \;\; y^* = (k^*)^{\,a} = 0.5839, \;\; c^* = y^* - \delta k^* = 0.4175$$

The quantitative calibration used to evaluate all solution methods is presented in Chapter 7, while the empirical results that compare the performance of these methods appear in Chapter 8.

Chapter 4. Numerical Solution Methods

In this chapter I give an overview of the model and the methods used throughout this thesis. The methods introduced here, Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM), and Neural Networks, are evaluated under the common calibration described in Chapter 7 and assessed using the performance metrics defined in Chapter 6. The results of this comparison are presented in Chapter 8.

4.1 Perturbation method

Perturbation method is one of the most widely used techniques when solving real business cycle models. It approximates the nonlinear equilibrium conditions of a RBC model by using a Taylor expansion around the deterministic steady state, allowing the representation of the model's policy functions as low order polynomial functions of the state variables.

 https://doi.org/10.26219/heal.aueb.9687

This approach comes with two key strong advantages. First, it is analytically trackable and computationally fast, since the approximated equations are expressed in closed forms and can be solved with standard linear algebra. This property stands particularly in simple RBC setups, such as the one sector framework I analyse on this thesis, where the dimensionality of the state space remains limited. Second, it preserves the structural features of the model, since the approximation is delivered from the model's equilibrium conditions, while accurately describing its local dynamics around the steady state. However, its main limitations are that it remains local. Accuracy decreases as shocks get bigger and push the system further from the steady state.

The analysis presented in this chapter for Perturbation method follows the general framework developed by Schmitt-Grohé & Uribe (2004), applied to a single agent real business cycle model described in chapter 3.

4.1.1 General formulation

To apply the Perturbation method, the model is expressed in the general state control representation:

$$\mathbb{E}f(y', y, x', x) = 0, \quad (4.1)$$

Where:

- $x \in \mathbb{R}^{n_x}$ is the vector of state variables, which are predetermined (capital and exogenous shocks)
- $y \in \mathbb{R}^{n_y}$ is the vector of control variables (consumption)
- And $f(\cdot)$ is a vector of equilibrium conditions implied by optimality and market clearing.

For the one sector RBC model:

$$x = \begin{bmatrix} k \\ logA \end{bmatrix}, \quad y = c, \quad (4.2)$$

And $f(\cdot)$ collects the Euler equation, the resource constraint and the low of motion for productivity:

$$\begin{cases} u'(c)\beta\mathbb{E}_t\left[u'(c')\left(aA'k'^{a-1} + 1 - \delta\right)\right] = 0 \\ c + k' - Ak^a - (1-\delta)k = 0 \\ \log A' - \rho \log A - \sigma\varepsilon' = 0 \end{cases}, \quad (4.3)$$

https://doi.org/10.26219/heal.aueb.9687

In the numerical implementation $f(\cdot)$ is coded in levels but evaluated using log-transformed variables for numerical stability, with

$$y = \ln c, \quad x_1 = \ln k, \ \ x_2 = \ln A$$

The Jacobian matrices $f_y, f_{y'}, f_x, f_{x'}$ and their higher order derivatives are computed symbolically at the steady state, as in Schmitt-Grohé and Uribe (2004).

4.1.2 Policy function representation

The equilibrium solution of the system can be presented as policy function that map the current state into next period outcomes

$$y = g(x, \sigma), \quad x' = h(x, \sigma) + \eta\varepsilon', \qquad (4.4)$$

Where $g(\cdot)$ and $h(\cdot)$ denote the policy and transition functions, respectively, and σ is the parameter that scales the volatility of the exogenous shocks.

At the deterministic steady state (x^*, y^*), uncertainty is absent with $\sigma = 0$, and:

$$f(y^*, y^*, x^*, x^*) = 0, \text{with } y^* = g(x^*, 0), \ x^* = h(x^*, 0) \quad (4.5)$$

The goal of the Perturbation method is the approximation of the $g(\cdot)$ and $h(\cdot)$ as Taylor series around $(x^*, \sigma = 0)$.

4.1.3 First Order Approximation

Expanding $g(x, \sigma)$ and $h(x, \sigma)$ to first order around the steady state $(x^*, 0)$ yields to:

$$g(x, \sigma) \approx y^* + g_x(x - x^*), \ h(x, \sigma) \approx x^* + h_x(x - x^*), \quad (4.6)$$

Where g_x and h_x are the matrices of partial derivatives evaluated at the steady state.

When substituting these linear approximations (4.6) into the equilibrium conditions (4.1) and differentiating with respect to x gives the first order system:

$$f_y g_x + f_{y'} g_x h_x + f_x + f_{x'} h_x = 0, \ \ (4.7)$$

Where all derivatives $\left(f_y, f_{y'}, f_x, f_{x'}\right)$ are evaluated at the steady state.

For the baseline specification $(\delta = 1, \rho = 0)$, the transition matrix is triangular, meaning that the second state variable $(\log A)$ is fully exogenous.

This is implemented in the code by setting the corresponding coefficients in h_x to zero:

$$h_x = \begin{bmatrix} h_{11} & h_{12} \\ 0 & 0 \end{bmatrix},$$

which implies that future productivity does not depend on current states and affects capital only through current period equations.

 https://doi.org/10.26219/heal.aueb.9687

Equation (4.7) defines a generalized eigenvalue problem. Solving it yields the linear decision rules for y and the stable transition dynamics for x:

$$y - y^* = g_x(x - x^*), \quad (4.8)$$

$$x' - x^* = h_x(x - x^*), \quad (4.9)$$

The first order Perturbation produces a certainly equivalent solution: the expected values of all endogenous variables equal their deterministic steady state values. In other words, the linear first order coefficients capture the first order sensitivity of the policies to deviations in capital and productivity. Therefore, it cannot capture risk effects or welfare changes that depend on higher moments.

4.1.4 Second order Approximation

To account for the effects of uncertainty and nonlinearity, the policy functions are expanded with quadratic terms to second order around the steady state:

$$g(x,\sigma) \approx y^* + g_x(x - x^*) + \frac{1}{2}(x - x^*)' g_{xx}(x - x^*) + \frac{1}{2} g_{\sigma\sigma}\sigma^2, \quad (4.10)$$

$$h(x,\sigma) \approx x^* + h_x(x - x^*) + \frac{1}{2}(x - x^*)' h_{xx}(x - x^*) + \frac{1}{2} h_{\sigma\sigma}\sigma^2, \quad (4.11)$$

Following Schmitt-Grohé and Uribe (2004) methodology, the coefficients $g_\sigma, h_\sigma, g_{x\sigma}$ and $h_{x\sigma}$ are zero at the steady state, implying that the linear and quadratic terms are independent of the volatility of the shocks. Hence, uncertainty affects only the constant term and its being transmitted to it through $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$.

Substituting equations (4.10) and (4.11) into equation (4.1) and matching terms of equal order in $(x - x^*)$ and σ gives a linear system that can be solved sequentially:

First, compute g_x and h_x from the first order system (4.7), then solve for the second derivatives g_{xx}, h_{xx} and the constant correction term $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$.

The resulting second-order approximation captures both the mean effects of uncertainty and the nonlinear responses of the model to larger shocks.

4.1.5 Economic Interpretation

At first order, the Perturbation method's solution yields a linear system equivalent to a log linearized real business cycle model. Shock affects deviations from the steady state proportionally, but the expected values of endogenous variables remain equal to their determinist levels.

https://doi.org/10.26219/heal.aueb.9687

At second order, the approximation allows asymmetric and risk sensitive dynamics. The constant terms $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ shift the mean of endogenous variables away from the steady state, capturing precautionary effects and welfare adjustments. Nonlinearities in g_{xx} and h_{xx} generate richer impulse responses that vary with shock size and persistence.

Thus, while the first order solution is appropriate for small and symmetric shocks, the second order approximation upgrades the depth of the analysis and makes it essential for studying welfare, risk premia and uncertainty effects.

In practice, higher order Perturbation solutions, (third or beyond) become computationally intensive because the number of derivative terms grows exponentially with the number of state variables. For this reason, up to second order approximations are used to solve small scale models like this one sector RBC. Large scale DSGE models are typically solved using first-order approximations, as the inclusion of many state variables makes higher order derivatives computationally expensive and increases the complexity of the solution space.

4.2 Value Function Iteration (VFI)

The Value Function Iteration (VFI) model provides a global numerical solution to dynamic economic models by directly applying the principle of dynamic programming. Unlike Perturbation techniques, which rely on local expansions around the steady state, Value Function Iteration (VFI) solves directly the model's Bellman equation on a grid of state variables, ensuring convergence to the true value function even far from equilibrium. That is also the reason why it's computationally intensive and more demanding. Thus, the structure of Value Function Iteration (VFI) and its mechanism make it the benchmark "ground truth" solution against other approximate methods. Throughout the calibrations that will follow, Value Function Iteration (VFI), will stand as the benchmark against the other methods that will be evaluated.

4.2.1 Bellman equation formulation

For the one sector RBC model, the representative household maximizes expected discounted utility:

$$V(k, A) = \max_{c,k'}\{u(c) + \beta\mathbb{E}[V(k', A')]\}, \quad (4.12)$$

Subject to the resource constraint:

$$c + k' = Ak^{a} + (1-\delta)k, \quad (4.13)$$

https://doi.org/10.26219/heal.aueb.9687

And the technology process:

$$\log A' = \rho \log A + \sigma \varepsilon', \quad (4.14)$$

4.2.2 Bellman to Euler equation connection

The Bellman equation (4.12) embodies the same optimality condition that gives rise to the Euler equation (4.3)

Taking the fist order condition of Bellman Equation with respect to k' yields:

$$u'(c) = \beta \mathbb{E}[V(k', A')], \quad (4.15)$$

While the envelope condition implies that:

$$V_k(k, A) = u'(c)(aAk^{a-1} + 1 - \delta). \quad (4.16)$$

Substituting the envelope condition (4.16) one period ahead into (4.15) gives:

$$u'(c) = \beta \mathbb{E}\left[u'(c')\left(aA'k'^{a-1} + 1 - \delta\right)\right], \quad (4.17),$$

Which is precisely the Euler Equation.

Hence, the Bellman and Euler equations are mathematically equivalent representations of the same intertemporal optimization problem.

The Bellman formulation expresses the recursive dynamic programming structure of the problem while the Euler equation expresses its first order optimality condition in terms of marginal utilities and returns.

4.2.3 Algorithm

The algorithm of Value Function Iteration (VFI) solves the Bellman equation (4.12) by iteratively applying the Bellman operator until the value function converges. The procedure ensures convergence to the unique fixed point of the dynamic programming problem under standard assumptions of concavity and discounting. The main steps are as follows:

1. Define the state and decision variables.

The state vector includes capital k and productivity A, while the decision variables are consumption c and next period capital k'. These variables are linked by the resource constraint (4.13).

2. Discretization of the state space.

 https://doi.org/10.26219/heal.aueb.9687

A grid for capital is constructed to cover the economically relevant range of values, ensuring sufficient density where the value function is more curved. Productivity A is represented by a discrete set of possible realizations of its stochastic process.

3. Initialize the value function.

Choose an initial value function guess $V^{(0)}(k, A)$, typically set to zero or to the utility of steady state consumption.

4. Value function update (Bellman operator).

For each grid point (k,A), evaluate the Bellman operator:

$$V^{(n+1)}(k, A) = \max_{c,k'}\{u(Ak^a - k') + \beta\mathbb{E}[V^{(n)}(k', A')]\}, \quad (4.18)$$

Subject to the feasibility constraint $Ak^a - k' \geq 0$.

This step computes the value of each feasible decision and retains the maximum, reflecting the agent's optimal choice.

5. Policy update.

Once the value function is updated, extract the corresponding optimal policy for capital:

$$k'^*(k, A) = \arg\max\{u(Ak^a - k') + \beta\mathbb{E}[V^{(n)}(k', A')]\}, \quad (4.19)$$

From which the optimal consumption policy follows as $c^*(k, A) = Ak^a - k'^*(k, A)$

6. Convergence criterion.

The iteration continues until the value function converges according to

$$\|V^{(n+1)} - V^{(n)}\|_\infty < \varepsilon_{eval}, \quad (4.20)$$

Where ε_{eval} is a small tolerance.

The Bellman operator is a construction mapping, guaranteeing convergence to the unique fixed point $V^*(k, A)$ and the corresponding optimal choices.

The algorithm yields the global solution $\{V^*(k, A), k'^*(k, A), c^*(k, A)\}$, which characterize the household's optimal dynamic behaviour.

https://doi.org/10.26219/heal.aueb.9687

Its step-by-step structure makes it straightforward to implement and serves as the basis for comparison with more advanced global methods introduced later, such as the Endogenous Grid Method (EGM) and Neural Network approximations.

4.2.4 Implementation Details

The implementation of the Value Function Iteration (VFI) follows the structure of the algorithm described above and adapts it to the baseline real business cycle (RBC) model. The purpose of this section is to describe the design logic of the numerical setup rather than the computational code itself.

The algorithm is performed over a discretized state space that covers the relevant range of capital and productivity levels observed in the model.

The grid for capital is chosen to be sufficiently dense to capture the curvature of the value and policy functions, with particular attention near the lower bound where constraints may bind. Since productivity is assumed to be independent and identically distributed, its stochastic process is represented by a small set of discrete realizations.

Expectations over future productivity are computed numerically by averaging the value function across possible states of nature.

In models with persistent shocks, this expectation would depend on the conditional probability of transition between productivity states; however, in the baseline calibration with i.i.d. shocks, it simplifies to a simple average.

At each iteration, the algorithm evaluates the Bellman operator on the defined grid by solving for the level of next period capital that maximizes the agent's lifetime utility subject to the resource constraint.

The iteration continues until the difference between two successive value functions falls below a pre-specified tolerance value, ensuring that both the value and policy functions have converged. This design ensures a globally accurate solution to the optimization problem, as the Bellman equation is evaluated across the entire state space rather than around a local approximation point.

While computationally demanding, this approach provides a reference benchmark for the other methods examined later, most notably the Endogenous Grid Method (EGM) and Neural

 https://doi.org/10.26219/heal.aueb.9687

Network (NN)–based approximations, which build upon the same problem structure but achieve higher computational efficiency.

4.2.5 Properties, Limitations, and Computational Aspects of Value Function Iteration (VFI)

The Value Function Iteration (VFI) method provides a globally accurate solution to dynamic programming problems by evaluating the Bellman equation over the entire state space. Its main advantage lies in its global convergence: under standard assumptions of concavity and discounting, the Bellman operator is a contraction mapping that guarantees convergence to the unique value function, regardless of the initial guess.

This property makes VFI particularly robust to large shocks, nonlinearities, and binding constraints, and therefore ideal for small-scale models such as the one-sector RBC framework.

However, this global accuracy comes at a significant computational cost. The method requires solving the maximization problem at every point in the discretized state space, a process that scales rapidly with the number of grid points. If Nk and NA denote the number of grid points for capital and productivity respectively, each iteration requires approximately $O(NkNA)$ evaluations of the Bellman operator. As the model becomes more complex—incorporating additional state variables such as labour, heterogeneous agents, or aggregate shocks—the number of required grid points increases exponentially, giving rise to the curse of dimensionality.

Several strategies can alleviate this burden. Vectorization and parallelization can reduce computational time, while interpolation and adaptive grids allow the use of fewer grid points without sacrificing accuracy. Alternative formulations, such as policy function iteration or Howard's policy improvement algorithm, can accelerate convergence by iterating on the policy rather than the value function. Functional approximation and projection methods can also reduce dimensionality by representing the value function with a smaller set of parameters instead of a full grid.

Even with such refinements, the computational requirements of VFI remain substantial for large-scale DSGE models. In those contexts, local Perturbation methods—though less accurate globally—offer a faster and more tractable solution.

https://doi.org/10.26219/heal.aueb.9687

For intermediate cases, where global accuracy is desired, but full VFI is infeasible, methods such as the Endogenous Grid Method (EGM) or Neural-Network-based approximations provide effective compromises between precision and efficiency.

4.3 Endogenous Grid Method (EGM)

The Endogenous Grid Method (EGM) offers a more efficient approach to solving dynamic stochastic models compared to the conventional Value function iteration (VFI). Endogenous Grid Method (EGM) was first developed and introduced by Carroll (2006) and further formalized by Maliar and Maliar (2014). The innovation of the Endogenous Grid Method (EGM) is the fact that can maintain the global accuracy of the Bellman framework while substantially reducing computational time.

More specifically, instead of performing a costly numerical maximization at each grid point, as Value Function Iteration (VFI), Endogenous Grid Method (EGM) constructs an endogenous grid of state variables that are consistent with agent's optimal decisions.

The inversion of the usual iteration process allows the model to be solved with fewer function evaluations and faster convergence, while preserving the accuracy of the Euler equation.

This method consists one of the standard tools in computational macroeconomics, when solving models of consumption savings and real business cycles.

In this thesis, Endogenous Grid Method (EGM) is applied to the same one sector real business cycle model introduced earlier, serving as an intermediate benchmark between the global accurate but computationally intensive Value Function Iteration (VFI) and the more flexible Neural Network approximations.

4.3.1 Theoretical Formulation

The Endogenous Grid Method (EGM) builds directly on the Euler equation, which characterizes optimal intertemporal allocation.

For the one sector real business model, the Euler equation (4.17) is:

$$u'(c) = \beta\mathbb{E}\big[u'(c')\big(aA'k'^{a-1} + 1 - \delta\big)\big], \quad (4.17)$$

Using this condition, the algorithm proceeds by solving for consumption as a function of next period capital and productivity, rather than performing direct maximization as in the Bellman framework.

 https://doi.org/10.26219/heal.aueb.9687

Substituting the resource constraint:

$$c = Ak^a + (1-\delta)k - k', \quad (4.21)$$

Into equation (30), we obtain an implicit relationship between c, k and k'.

For each possible value of k' and A, the Euler equation (4.17) can be used to compute the corresponding current consumption c.

The latter defines an endogenous grid for capital, denoted by $k = G(k', A)$. Unlike in the Value Function Iteration (VFI) method, where the grid for k is fixed exogenously and optimisation determines the next period choice k', the Endogenous Grid Method (EGM) reverses this logic. It begins with a grid of k' and, using the Euler equation (4.17), computes the set of current capital values k that make those future choices optimal. The resulting grid is therefore endogenous. It arises from the model's first order conditions rather than being imposed priori.

4.3.2 Algorithm

The Endogenous Grid Method (EGM) reverses the order of computation used in the Value Function Iteration (VFI) procedure. Whereas Value Function Iteration (VFI) fixes the grid of current capital k and searches for the optimal next-period value k' through numerical maximization, Endogenous Grid Method (EGM) fixes a grid for k' and derives the corresponding optimal k that satisfies the Euler equation (4.17). This inversion eliminates the need for repeated optimization and thus reduces computational cost while maintaining exact satisfaction of the model's first-order condition.

The algorithm proceeds as follows:

1. Specify a grid for next period capital k'.

   Choose a set of values for k' covering the economically relevant range. If productivity A is stochastic, include discrete realizations of its process. This grid defines the endogenous decision variable of the problem.

2. Compute next period consumption and marginal utility.

   For each grid point (k',A), evaluate next period consumption from the resource constraint (4.21) and derive the corresponding marginal utility $u'(c)$.

3. Apply the Euler equation to recover current consumption c.

   Using

https://doi.org/10.26219/heal.aueb.9687

$$u'(c) = \beta \mathbb{E}[u'(c')R'], \quad R' = aA'k'^{a-1} + 1 - \delta, \quad (4.22)$$

compute the current marginal utility and invert $u'(\cdot)$ to obtain current consumption c. This step enforces the optimal intertemporal condition exactly rather than approximately.

4. Construct the endogenous grid.
   Combine each computed c with the resource constraint, equation (3.4)

$$c + k' = y + (1 - \delta)k, \quad (3.4)$$

   to recover the implied current capital level k. The resulting pairs (k,c) form the endogenous grid consistent with the agent's optimal behaviour.
5. Interpolate to a standard grid for current capital.
   Interpolate the computed pairs onto an evenly spaced grid for k to obtain a continuous consumption policy c(k,A). This ensures comparability with the Value Function Iteration (VFI) benchmark and facilitates simulation.
6. Update and iterate.
   Use the new consumption policy to compute updated expectations and repeat the procedure until the policy function changes by less than a pre-specified tolerance. Because EGM directly satisfies the Euler equation, convergence is typically reached in only a few iterations.

By deriving the optimal mapping between current and future capital directly from the Euler condition, Endogenous Grid Method (EGM) circumvents the numerical maximization step that dominates the computational cost in Value Function Iteration (VFI). This efficiency gain makes it possible to achieve global accuracy on much finer grids or to extend the analysis to models with additional stochastic dimensions at a manageable computational cost.

4.3.3 Implementation Details

The implementation follows the same real business cycle structure and parameterization as the Value Function Iteration (VFI) benchmark. The model is solved in log transformed variables for numerical stability, and the utility and production functions retain the same forms:

$$u(c) = \frac{c^{1-\gamma} - 1}{1 - \gamma}, \quad f(k, A) = Ak^a, \quad (4.23)$$

The algorithm begins by specifying the next period's capital grid, which becomes the main endogenous variable of iteration. At each grid point, the Euler equation is used to compute the

 https://doi.org/10.26219/heal.aueb.9687

implied current consumption c and its corresponding level of current capital k. This produces a set of endogenous points (k,c), which are interpolated to form a continuous consumption policy function. The policy is then updated through forward simulations until the implied consumption c and savings choices k' stabilize.

Expectations over productivity A are evaluated numerically. In the baseline calibration with independent and identically distributed shocks (ρ=0), expectations reduce to unconditional averages. For persistent processes, integration would instead occur over the conditional distribution of A'. The algorithm continues until the consumption policy function changes by less than a small tolerance, which indicates convergence.

4.3.4 Properties and computational advantages

The Endogenous Grid Method (EGM) retains the global convergence and accuracy of the Value Function Iteration (VFI) method, while achieving substantial computational savings. By constructing the endogenous grid directly from the Euler equation (4.17), it avoids the computational heavy maximization step inherent in Value Function Iteration (VFI) and reduces the number of function evaluations by several orders of magnitude. Therefore, the Endogenous Grid Method (EGM) scales efficiently even when the grid is dense, making it particularly suitable for high precision applications or models requiring frequent re-solutions, such as estimation or policy analysis.

In contrast to Perturbation method, which rely on local approximations, the Endogenous Grid Method (EGM) provides a global solution that respects all nonlinear properties in the model. At the same time, it is considerably faster than Value Function Iteration (VFI) because the computational cost grows approximately linearly with the number of grid points rather than exponentially.

The main limitation of Endogenous Grid Method (EGM) arises in models with nonconvexities or occasionally binding constraints, where the inversion of the Euler equation (30) may no longer yield a monotonic mapping between states and controls. In such cases, hybrid methods combining Endogenous Grid Method (EGM) with localized maximization or Neural Network based function approximation can be employed.

The Endogenous Grid Method (EGM) provides an elegant balance between the theoretical precision of dynamic programming and the computational efficiency of direct methods. It produces results that are nearly identical to those of Value Function Iteration (VFI) while

https://doi.org/10.26219/heal.aueb.9687

requiring only a fraction of the computational time. For this reason, it has become the preferred global solver for one-dimensional stochastic growth and consumption–savings problems.

4.4 Neural Networks (NN)

Recent advances in computational macroeconomics have demonstrated that machine learning techniques, particularly artificial Neural Networks, consist one of the powerful tools for solving dynamic stochastic general equilibrium and real business cycle models. Neural Networks offer an alternative to conventional numerical methods such as Perturbation, Value Function Iteration (VFI) and the Endogenous Grid Method (EGM) by providing a flexible functional approximation that captures complex nonlinear relationships without relying on predefined grids or specified functional forms.

The motivation for applying Neural Networks in this context arises from a simple observation. The equations that characterize the dynamic economic models, typically Euler, Bellman and recourse constraints, define mappings between states and optimal decisions.

Traditional methods approximate these mapping on discrete grids or through local polynomial expansions, as was discussed on previous chapters. Neural Networks, in contrast, can approximate them globally and continuously, using adaptive parameters that are estimated by minimizing residual errors from the model's equilibrium conditions. The latter allows the solution to scale efficiently to higher dimensions and to capture strong nonlinearities that would otherwise require prohibitively dense grids.

4.4.1 What is a Neural Network?

A Neural Network is a computational model, inspired by the structure of the human brain, designed to approximate unknown functions from data or from a system of equations. In its simplest form, a feedforward Neural Network consists of layers of interconnected nodes, also known as neurons, each performing a weighted sum of its inputs followed by a nonlinear transformation known as activation function.

Formally, for an input vector $x \in \mathbb{R}^n$, a network with one hidden layer produces the output:

$$\hat{y} = f(x;\theta) = \varphi_2(W_2\varphi_1(W_1x + b_1) + b_2), \qquad (4.24)$$

Where W_1 and W_2 are matrices of weights, b_1 and b_2 are bias vectors, $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ are nonlinear activation functions (such as ReLU, tanh, or sigmoid), and $\theta = \{W_1, W_2, b_1, b_2\}$ denotes the set of all network parameters.

https://doi.org/10.26219/heal.aueb.9687

By adjusting these parameters trough a training process, the Neural Network learns to approximate a target function, based on observed examples or, in the context of this thesis, from simulated model equations. Each hidden layer adds representational capacity, allowing the Neural Network to approximate increasingly complex functions.

According to the Universal Approximation Theorem, even a single hidden layer network can approximate any continuous function on a compact domain, provided it has a sufficient number of neurons. In practice, deeper networks with multiple hidden layers often achieve better accuracy and generalization, as they can represent hierarchical relationships between inputs and outputs.

4.4.2 Why Neural Networks are suitable for solving Economic models

In dynamic macroeconomic models, agent's optimal decisions, such as consumption and capital accumulation, can be represented as unknown functions of state variables $c = g(k, A)$ and $k' = h(k, A)$.

The challenge lies in accurately approximating these policy functions under uncertainty, nonlinearity and sometimes high dimensionality.

Neural Networks are particularly suitable for this task for several reasons:

1. Global functional flexibility.
   Neural Networks can approximate arbitrary nonlinear mappings without requiring a specific functional form. This makes them capable of capturing nonlinear policy functions and non-covex decision rules that may arise in complex DSGE environments.
2. Dimensional scalability.
   Traditional grid-based methods (Value Function Iteration (VFI), Endogenous Grid Method (EGM)) suffer from the curse of dimensionality, as computational cost increases exponentially with the number of state variables. Neural Networks, on the other hand, scale linearly with the number of parameters and can efficiently handle multi-dimensional state spaces.
3. Smoothness and differentiability.
   The use of smooth activation functions ensures that the approximated policy functions are continuous and differentiable. This property is valuable for evaluating derivatives or simulating impulse responses, which are central to macroeconomic analysis.
4. Data driven learning.

https://doi.org/10.26219/heal.aueb.9687

Instead of discretizing the state space, the network learns directly from simulated samples of the model's equilibrium conditions. The latter allows for flexible sampling strategies, such as focusing on regions with high probability mass or strong nonlinear responses, improving efficiency and accuracy.

By combining these four properties, Neural Networks provide a unified framework that can approximate value functions, policy functions, or even entire equilibrium mappings without discounting accuracy. Therefore, Neural Networks can represent a global, mesh-free alternative to classical numerical techniques and are particularly useful for models that are either high dimensional or face strong nonlinearities.

4.4.3 Neural Networks in the context of the real business cycle model

In this thesis, Neural Networks are employed to approximate the policy functions of the same one sector real business cycle model analysed in the previous chapters. The model's structure, defined by the Euler equation (4.17), resource constraint (3.4), and stochastic productivity process (3.6), remains unchanged.

However, instead of solving the system on a discrete grid (as in Value Function Iteration (VFI) and Endogenous Grid Method (EGM)), the policy functions are represented by parametric Neural Networks, whose parameters are estimated to minimize the Euler residuals across simulated data points.

4.4.4 Neural Network Architectures Implemented

Two different architectures are implementation are developed to approximate the policy function of the Real Business Cycle model. Both architectures share the same feedforward structure but differ in their source of learning signal and training strategy:

1. NN (Global) a distillation plus refinement model that learns initially from the Endogenous Grid Method (EGM) policy and is subsequently fine-tuned using Euler equation residuals.
2. NN (Equation) a fully independent model trained solely from the model's structural equation, without relying on any precomputed policy to mimic.

Both Neural Network based models approximate the policy mapping $G(k, A) = k'$, which defines next period capital as a function of current capital k and productivity A.

 https://doi.org/10.26219/heal.aueb.9687

4.4.4.1 NN (Global): Distillation and Euler Refinement

The first network, denoted NN(Global), combines supervised learning with model-based optimization. The training process proceeds in two stages:

1. Distillation stage:
   The network learns from the high precision and extremely fast Endogenous Grid Method (EGM) solution, which serves as a teacher policy $G^{EGM}(k, A)$.
   $$\mathcal{L}_{distill} = \mathbb{E}\left[\left(k'_{NN}(k, A) - k'_{EGM}(k, A)\right)^2\right] \qquad (4.25),$$
   A mixed training dataset is constructed:
   - 50% economically correlated samples, generated by simulating the true model dynamics under stochastic productivity ensuring that the network learns within the ergodic region of the economy,
   - 50% uniformly sampled points, to guarantee adequate coverage of the state space. The network is trained to minimize the mean squared error (MSE) between its predicted next-period capital k'_θ and the Endogenous Grid Method (EGM) benchmark values.
2. Euler refinement stage:
   After distillation, the model is fine-tuned by minimizing the Euler equation residual loss:
   $$\mathcal{L}_{Euler} = \mathbb{E}\left[\left(1 - \frac{\beta\mathbb{E}[u'(c')R']}{u'(c)}\right)^2\right] \qquad (4.26),$$
   Where the expectations are computed using Gauss-Hermite quadrature over future productivity shocks. This step enforces exact intertemporal optimality and removes any remaining approximation bias from the distillation phase.

For each state, the network outputs a saving share $s_\theta(k, A) \in (0,1)$, determining next-period capital as:

$$k' = s_\theta(k, A)Ak^a \qquad (4.27).$$

The resulting policy function $G_\theta^{Global}(k, A)$ combines stability of data driven imitation with the precision of equation-based learning. Empirically, this hybrid approach converges quickly and reproduces the benchmark policy with high accuracy, as verified through Euler residual diagnostic on calibration chapter.

4.4.4.2 NN (Equation): Pure Equation-Based Learning

The second architecture, NN (Equation), is trained entirely from the model's structural conditions, without any reference to pre-solved policies, as NN(Global), such as Endogenous Grid Method (EGM) or Value Function Iteration (VFI).

The learning objective directly enforces the Euler equation at each sampled point (k,A).

Training data are again composed of:

- 50% correlated economic trajectories, simulated under the stochastic process of productivity, and
- 50% uniformly distributed points across the state space.

For each state, the network outputs a saving share $s_\theta(k, A) \in (0,1)$, determining next-period capital as:

$$k' = s_\theta(k, A)Ak^a \quad (4.27).$$

The model parameters are optimized by minimizing the squared Euler residual:

$$\mathcal{L}_{Euler} = \mathbb{E}\left[\left(1 - \frac{\beta\mathbb{E}[u'(c')R']}{u'(c)}\right)^2\right] \quad (4.26),$$

Where consumption is defined by $c = Ak^a - k'$.

Since no external guidance is used, this network learns the optimal policy purely from first principles — a self-contained solver analogous to the "end-to-end" frameworks proposed by Maliar and Maliar (2021). While computationally heavier to train, the NN(Equation) demonstrates the capability of Neural Networks to discover globally consistent solutions directly from equilibrium conditions, confirming the potential of machine-learning-based solvers for macroeconomic models.

4.4.5 Architecture Design and Training Configuration

Both networks share the same parametric form:

- Input layer: 2 neurons
- Hidden layers: 3 fully connected layers, each with 64 neurons and tanh activations
- Output layer: 1 neuron, producing bounded saving share $s_\theta(k, A)$ via a scaled sigmoid transformation ensuring $s_\theta \in (0,1)$

https://doi.org/10.26219/heal.aueb.9687

- Temperature parameter $T = 2.5$ for output scaling and small boundary margin $\varepsilon = 10^{-4}$ to avoid corner solutions

Optimization is performed using the Adam algorithm with learning rate 10^{-3}, batch size 2048, and total training epochs of approximately 2200–2500, depending on the variant.

4.4.6 Comparative Overview

Feature	NN (Global)	NN (Equation)
Learning source	Distillation from EGM + Euler fine-tuning	Pure Euler-equation residual minimization
Dependence on other solvers	Requires EGM as teacher	Fully independent
Data generation	50% economic, 50% uniform	50% economic, 50% uniform
Objective function	MSE (distillation) + Euler residual loss	Euler residual loss only
Training epochs	800 + 1400	2500
Accuracy target	Mimic EGM and satisfy Euler	Satisfy Euler from scratch
Interpretation	Hybrid “teacher–student” model	Self-contained equation-based solver

4.4.7 Conclusion

The Neural Network approaches presented on this chapter extend traditional numerical methods by replacing discrete optimization and grid-based interpolation with continuous, differentiable function approximations. The NN (Global) model illustrates how hybrid learning can leverage existing solvers, such as Endogenous Grid Method (EGM), to accelerate convergence and improve stability. In contrast, the NN (Equation) demonstrates that the Euler condition alone is sufficient to identify and recover the optimal policy without external guidance.

https://doi.org/10.26219/heal.aueb.9687

Although the NN (Global) model benefits from its mimic behaviour of a teacher policy, this dependency also implies that in more complex settings, where the Endogenous Grid Method (EGM), or other teacher policy methods struggle, the Neural Network may inherit this liability and will magnify the approximation errors. To address this limitation, the NN (Equation) was developed as a fully independent solver, sacrificing computational efficiency for methodological robustness and autonomy from any teacher.

Empirically (see chapter 7. Calibration), both networks achieve near-benchmark accuracy while dramatically reducing the need for dense grids. The latter demonstrates that Neural Networks can act as global mesh-free solvers for dynamic stochastic general equilibrium models, capable of approximating complex nonlinear policies within the same theoretical consistency as conventional methods.

Chapter 5. From Local to Global solution methods: Stress Testing the Perturbation approach

Perturbation method, as discussed in chapter 4, offer an elegant and computationally efficient way to approximate the equilibrium dynamics of real business cycle models. By linearising or quadratically expanding the equilibrium conditions around the deterministic steady state, Perturbation methods provide analytical local solutions. However, because they rely on Taylor expansions, their accuracy is inherently local. If we move further away from the steady state, either due to large shocks or high volatility, the approximation quality deteriorates.

To determine the effective and acceptable range of reliability of Perturbation methods, under the predetermined model settings (Schmitt-Grohé and Uribe, 2004), in this chapter I perform a systematic stress test of first and second order Perturbation solutions against a global Value Function Iteration (VFI) benchmark, in order to answer the following question: Which is the point where Perturbation methods fail approximating accurately the model's dynamics under these settings?

5.1 Stress Test design

The analysis evaluates two local solutions

- Perturbation, 1st order: first order (linear) approximation around the steady state
- Perturbation, 2nd order: second order (quadratic) approximation in logarithms

Each is compared against a globally accurate Value Function Iteration (VFI) policy that solves the Bellman equation () on a refined volatility-adapted grid.

The comparison is carried out across a range of volatility values ($\sigma \in [0,0.35]$) and for a spectrum of shock sizes ε applied to productivity. For each configuration, the algorithm checks whether the Perturbation -based policy satisfies three diagnostic categories relative to the Value Function Iteration (VFI) benchmark.

1. Euler optimality, unit free and expressed in log10 scale
2. Policy proximity, RMSE and sup-norm distance of k' relative to the Value Function Iteration (VFI) benchmark
3. Impulse response closeness, absolute deviation (in % of steady state) and relative deviation (in % of VFI IRF amplitude) for one off productivity shock.

5.2 Accuracy regimes

To interpret the results consistently, two accuracy regimes are defined.

Criterion	**Set A — Certified**	**Set B — Operational**
Euler residuals	p95 ≤ -2.0, max ≤ -1.8	p95 ≤ -1.5, max ≤ -1.2
Policy distance	RMSE $\leq 10^{-3}$, sup $\leq 5\times10^{-3}$	RMSE $\leq 5\times10^{-3}$, sup $\leq 2\times10^{-2}$
IRF deviations	$\lvert\Delta k\rvert \leq 1.5\%$, $\lvert\Delta c\rvert \leq 1.0\%$, rel $\leq 10\%$	$\lvert\Delta k\rvert \leq 2.0\%$, $\lvert\Delta c\rvert \leq 1.5\%$, rel $\leq 20\%$

Only when all three criteria are satisfied simultaneously is the Perturbation method considered reliable for that configuration. For each σ, a bisection search identifies the largest shock ε* that meets all tests. The resulting function ε*(σ) defines the range where Perturbation is valid under each accuracy regime.

https://doi.org/10.26219/heal.aueb.9687

5.3 Diagnostic results

The results of the stress test are summarised in Figure 5.1, which reports the largest one-off productivity shock ε*(σ) that satisfies all diagnostic gates for both first- and second-order Perturbation methods under the Certified (Set A) and Operational (Set B) accuracy regimes. The figure demonstrates a non-monotonic, hump-shaped pattern of reliability.

Under Set A (Certified), ε* increases up to 6.5–9.5% for σ=0.03–0.05 but collapses again to 0.01% for $\sigma \geq 0.07$.

Under Set B (Operational), Perturbation remains valid for 5–12.5% one-off shocks for σ=0.02–0.07, while at both very low and very high volatility, the admissible range shrinks to 0.10%.

This behaviour highlights how Perturbation accuracy is maximised at moderate volatility levels, where the state space remains sufficiently close to the deterministic steady state, but nontrivial curvature is still captured. At the extremes—either near-zero or high volatility—accuracy deteriorates: at low σ, the relative IRF criteria become excessively restrictive, and at high σ, nonlinearities and approximation errors accumulate rapidly.

These results create a non-monotonic, hump-shaped pattern of reliability: Perturbation performs best at moderate σ (0.02–0.07) and becomes unreliable both when volatility is extremely low or high.

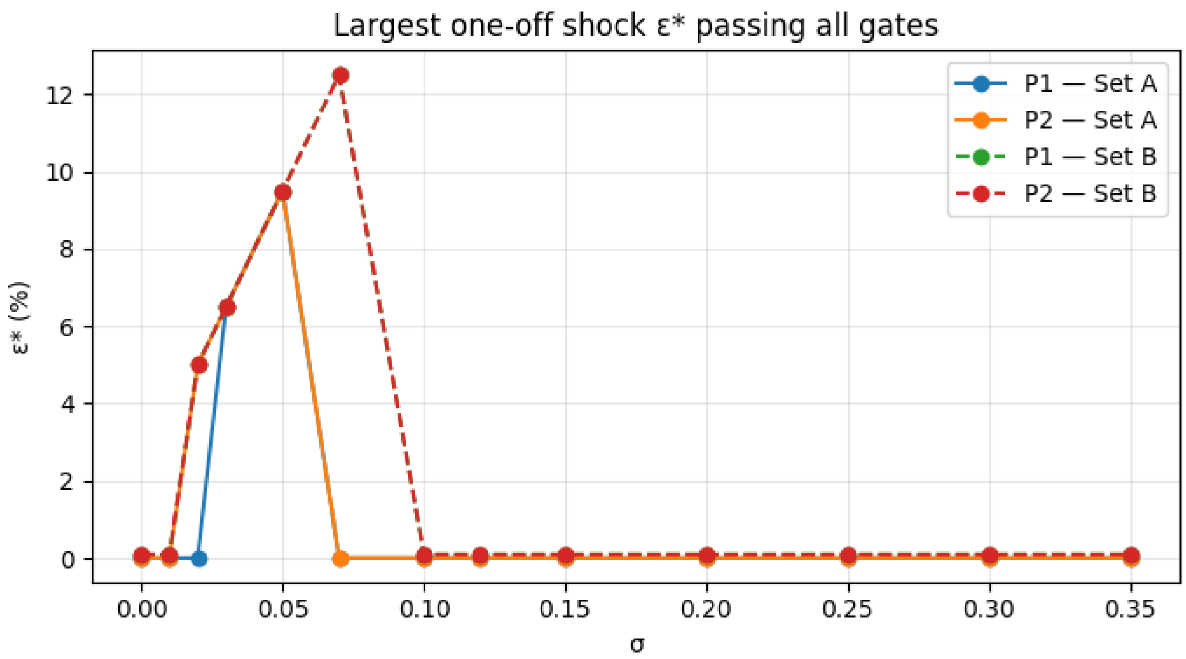


Figure 5.1: Valid Shock Size Boundary for Perturbation Methods

The figure shows the largest one-off productivity shock ε(σ) that satisfies all diagnostic accuracy criteria for first-order (P1) and second-order (P2) Perturbation under both Certified

 https://doi.org/10.26219/heal.aueb.9687

(Set A) and Operational (Set B) regimes. The hump-shaped pattern between σ=0.03 and 0.07 indicates the region of maximum reliability for local approximation methods.

5.4 Certified shock size

Aggregating these results yields a clear quantitative benchmark:

> With ρ = 0 and the stated thresholds, both first- and second-order Perturbation methods are reliable for one-time productivity shocks up to approximately 0.01% across the tested range of σ. Within moderate volatility ($\sigma \in [0.03, 0.05]$), the same strict criteria allow shocks of 6.5–9.5% before any diagnostic gate is breached.

This certified safe shock size, ε*=0.01%, is deliberately conservative. It ensures that local optimality, policy accuracy, and impulse response dynamics remain numerically indistinguishable from those obtained with a fully nonlinear global solution. The stress-test framework therefore defines a quantitative boundary for the region where local Perturbation approximations are guaranteed to be valid under strict accuracy criteria.

The numerical thresholds adopted here (Euler p95 ≤ −2.0, policy RMSE ≤ 10^{-3}, IRF deviations ≤ 1–1.5%) are intentionally stringent to achieve perfect, machine-level equivalence with the global benchmark. However, in practical macroeconomic applications, slightly larger deviations are acceptable provided that the impulse responses remain visually aligned with those from a global method.

Under this interpretation, and consistent with the present calibration:

- Low volatility (σ ≤ 0.03): Perturbation is accurate for 5–10% one-off shocks, and fully reliable under both orders.
- Moderate volatility (σ ≈ 0.05–0.07): residuals remain small, but IRF alignment becomes sensitive; shocks up to 12.5% remain acceptable under Set B, whereas Set A drops to 0.01%.
- High volatility (σ ≥ 0.10): deviations become substantial, reducing the valid region to ≤0.1%, and Perturbation should be replaced by a global solver.

These findings provide a practical and transparent decision:

> use Perturbation for local and low-volatility analysis and validate results with a global method for larger shocks or when volatility exceeds σ ≈ 0.07.

https://doi.org/10.26219/heal.aueb.9687

5.5 Why switch to global methods

While Perturbation is analytically elegant and computationally fast, its reliability is fundamentally local. The stress test demonstrates that once the economy moves sufficiently far from the deterministic steady state—whether due to larger shocks, higher volatility, or persistence—the approximation errors in Euler equations and policy functions rise sharply. Global methods such as Value Function Iteration (VFI), Endogenous Grid Method (EGM), or Neural Network–based solvers compute policy functions directly on the full state space and maintain accuracy under nonlinear dynamics.

From a practical perspective:

- Local radius of validity: Perturbation performs exceptionally near the steady state but cannot capture off-equilibrium curvature or regime shifts.
- Policy safety: For quantitative or policy exercises involving large disturbances, global methods ensure error control and robustness.
- Recommended workflow: Employ Perturbation for calibration and sensitivity analysis but confirm key findings with a global solver whenever shocks exceed the indicated operational thresholds.

5.6 Conclusion

This analysis quantifies the boundary of reliability for Perturbation methods within the baseline RBC model. Under the current calibration (α=0.30,β=0.95,γ=2,ρ=0):

Perturbation performs exceptionally well for moderate volatility ($\sigma \in [0.02,0.07]$), where it remains accurate for 5–12.5% shocks (Set B) and up to 6.5–9.5% under the strict certification (Set A). For both very low and very high volatility ($\sigma \leq 0.01$ or $\sigma \geq 0.10$), the safe range collapses to ≤ 0.1%, requiring the use of global solvers.

This stress-test therefore establishes a quantitative threshold separating the domain of local validity from the domain requiring global numerical methods.

Figure 5.4. Largest one-off productivity shock $\varepsilon*(\sigma)$ that satisfies all diagnostic gates for first- (P1) and second order (P2) Perturbation under Set A (Certified) and Set B (Operational). The hump around σ=0.03–0.05 and the collapse for $\sigma \geq 0.10$ reflect, respectively, binding IRF constraints at very low volatility and nonlinear amplification at high volatility. Having established the boundary of local validity, Chapter 8 evaluates how global solvers perform across the same volatility range.

https://doi.org/10.26219/heal.aueb.9687

Chapter 6. Performance Metrics

To evaluate and compare the performance of different solution methods developed in this thesis (Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM) and Neural Network (NN)-based solutions), a set of consistent quantitative metrics is required. These assess both numerical accuracy and the computational efficiency of each method, allowing for a structured evaluation of their relative strengths and limitations.

The metrics employed in the calibration and validation exercises fall into four main categories:

1. Euler residuals (unit-free measures of equilibrium accuracy),
2. Policy distance metrics (RMSE and sup-norm errors),
3. Impulse response function (IRF) errors (absolute and relative deviations), and
4. Runtime and computational efficiency

Each metric captures a distinct dimension of model performance and, collectively, they provide a comprehensive framework for assessing the reliability of each numerical solution.

6.1 Euler residuals

The Euler equation residual quantifies the degree to which a computed policy satisfies the model's first order optimality condition. For the representative household in the Real Business Cycle model, optimality implies:

$$1 = \beta \mathbb{E}_t \left[\frac{u'(c')R'}{u'(c)} \right] \qquad (6.1),$$

where $R' = aA'k'^{a-1} + 1 - \delta$, and $u'(c)$ denotes the marginal utility of consumption.

The unit-free Euler residual is defined as:

$$EE(k, A) = \left| 1 - \frac{\beta \mathbb{E}_t[u'(c')R']}{u'(c)} \right| \qquad (6.2).$$

This measure expresses the relative deviation from the exact Euler condition. In a perfectly solved model $EE(k, A) = 0$; in numerical approximations, the magnitude of $EE(k, A)$ indicates the residual error in the interpolar condition. To summarise performance, the statistics $\log 10\ (EE)$ median, 95th percentile, and maximum are computed on economically correlated (ergodic) samples generated by each method's policy, with additional uniform coverage where noted, to ensure relevance across the state space.

https://doi.org/10.26219/heal.aueb.9687

A typical interpretation is:

$$\log 10\,(EE) = -4 \;\Rightarrow EE = 10^{-4},$$

Which corresponds to a relative Euler-equation error of 0.01% at the grid point. Such small values indicate that the computed policy satisfies near-perfect intertemporal optimality.

In numerical implementation, a slightly regularised version of the expression is used to avoid division by very small marginal utilities. Specifically, the denominator may include a stabilising term, e.g. $1 + |u'(c)|$, without altering the interpretation of the residual.

6.2 Policy distance metrics

6.2.1 Root mean square error (RMSE)

Policy RMSE quantifies the average deviation of the computed policy function from the reference (benchmark) policy across the state space:

$$RMSE(k', A) = \sqrt{\mathbb{E}[(k'_{method} - k'_{bechmark})^2]} \quad (6.3).$$

In this thesis, for the calibrations, the benchmark is the globally accurate Value Function Iteration (VFI) policy, and the alternative is the policy generated by the Perturbation, Endogenous Grid Method (EGM), or a Neural Network solver.

6.2.2 Sup-norm (Maximum absolute error)

The sup-norm (or maximum absolute deviation) complements the RMSE by capturing the worst-case discrepancy:

$$sup - error(k', A) = \max_{(k,A)} |k'_{method} - k'_{bechmark}| \qquad (6.4).$$

This metric is particularly useful for identifying whether the approximation errors are concentrated locally or spread across the state space. Together, RMSE and sup-error assess the pointwise accuracy of policy functions under different solvers.

6.3 Impulse response Function (IRF) errors

Impulse response Function (IRFs) describe the transition dynamics of the under-investigation economy following a one-off productivity shock. For each method, IRFs of key variables such as capital and consumption are computed and compared to those generated by the benchmark (Value Function Iteration (VFI) solution).

https://doi.org/10.26219/heal.aueb.9687

Two complementary error measures are used:

6.3.1 Absolute IRF error

$$|\Delta x|_{abs} = 100 \times \frac{max_t\left|x_t^{method} - x_t^{benchmark}\right|}{x^*} \qquad (6.5),$$

Where $x \in \{k, c\}$ and x^* denotes the steady state value.

This measures the maximum deviation as a percentage of the steady state and indicates the magnitude of response mismatch.

6.3.2 Relative IRF error

$$|\Delta x|_{rel} = 100 \times \frac{max_t\left|x_t^{method} - x_t^{benchmark}\right|}{max_t\left|x_t^{benchmark} - x^*\right|} \qquad (6.6),$$

This expresses the deviation relative to the amplitude of the benchmark IRF, thus isolating distortions in dynamic shape rather than level.

Both IRF metrics assess the dynamic accuracy of a solver in reproducing the economy's response to shocks.

6.4 Runtime and computational Efficiency

Beyond accuracy, the performance of a numerical method depends critically on its computational cost. For each solver, total runtime, number of iterations and convergence speed are recorded.

Efficiency is evaluated through:

1. Execution time (seconds): total wall-clock time required for convergence
2. Iterations to convergence: number of policy or value iterations before satisfying tolerance criteria
3. Grid or network complexity: numbers of nodes, Gauss Hermite points, or neural parameters required for convergence.

A method is considered efficient if it attains comparable accuracy metrics (Euler, Bellman, policy, IRF) at a fraction of the computational cost. Runtime analysis thus complements accuracy diagnostics by identifying the trade-off between precision and efficiency.

https://doi.org/10.26219/heal.aueb.9687

6.5 Summary of Evaluation Framework

The metrics presented above form the quantitative foundation for comparing solution methods in the calibration chapter. They allow each solver, Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM), and Neural Network, to be assessed along two complementary dimensions:

1. Accuracy: how closely the numerical solution satisfies optimality conditions and replicates the benchmark dynamics.
2. Efficiency: how quickly and economically the method attains that accuracy.

Table 6.1 summarizes the key metrics and their interpretation.

Metric	Definition / Description	Interpretation
Euler residual ($\log_{10}$)	Unit-free deviation from intertemporal optimality	Measures local accuracy of FOCs
Policy RMSE	Mean-squared policy difference vs. benchmark	Average accuracy of decision rule
Policy sup-error	Maximum deviation vs. benchmark policy	Worst-case error in decision rule
IRF abs. error	Max deviation (% of steady state)	Magnitude of dynamic mismatch
IRF rel. error	Max deviation (% of benchmark amplitude)	Shape difference in dynamic response
Runtime / iterations	Execution time and convergence steps	Computational efficiency and scalability

6.6 Conclusion

This chapter has defined the set of performance metrics used to evaluate the reliability and efficiency of numerical solution methods in dynamic stochastic models. By quantifying equilibrium accuracy, policy deviations, dynamic responses, and computational efficiency,

 https://doi.org/10.26219/heal.aueb.9687

these metrics provide a consistent and multidimensional basis for comparison. The next chapter introduces the unified calibration under which all methods, Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM), and Neural Networks (NN) are solved, before Chapter 8 applies the metrics developed here to assess and compare their performance.

Chapter 7. Calibration

This chapter defines the quantitative calibration adopted throughout the analysis. All numerical methods, Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM), and Neural Networks (NN), are evaluated under a common baseline setup derived from the canonical one-sector Real Business Cycle (RBC) model.

This calibration follows the benchmark setup of Schmitt-Grohé and Uribe (2004), ensuring comparability with previous studies while preserving interpretability across both local and global solution methods.

7.1 Baseline model and Parameters

The economy is characterized by a representative agent who maximizes expected lifetime utility under stochastic productivity as described in Chapter 3. The parameters used in the calibration are reported in Table 7.1.

Parameter	**Symbol**	**Value**	**Description**
Capital share	α	0.30	Implies ~5% annual real interest rate
Discount factor	β	0.95	Matches OECD factor income shares
Risk aversion	γ	2.00	Standard macroeconomic range (1–3)
Depreciation rate	δ	1.00	Full depreciation (baseline)
Shock persistence	ρ	0.00	i.i.d. shocks for stress test
Shock volatility	σ	$\sigma \in [0.005, 0.2]$	Stress test range

The chosen parameter values are standard in the Real Business Cycle literature and are consistent with empirical estimates for advanced economies.

https://doi.org/10.26219/heal.aueb.9687

A discount factor of β=0.95 implies an annual real interest rate close to 5%, while a capital share of α=0.3 matches average factor income shares observed in OECD data. The degree of relative risk aversion γ=2 lies within the conventional macroeconomic range of 1–3. Given $\rho = 0$, productivity shocks are independently and identically distributed (i.i.d.), which simplifies the stochastic environment and allows a clean isolation of volatility effects. Volatility σ is varied from 0.005 to 0.20 as part of the stress test.

This configuration follows the logic of a volatility–shock stress test, where the models introduced in Chapter 4 are solved repeatedly for different values of σ to examine their performances under increasing uncertainty.

7.2 Numerical Grids and State Space

Since all numerical methods, Value Function Iteration (VFI), Endogenous Grid Method (EGM), Perturbation, and Neural Networks, require evaluating functions over a discrete set of states, the accuracy of each solution critically depends on the design of the computational grid. To ensure comparability, a common grid structure is defined for the endogenous state k and the exogenous shock A, with ranges that adapt to the volatility parameter σ.

Capital and productivity are both defined on evenly spaced grids, whose range expands with σ to ensure sufficient coverage of the relevant economic states. For each configuration, the Gauss–Hermite quadrature with $n_{GH} = 7 - 15$ nodes are used to approximate expectations with respect to the log-normal productivity distribution.

The grid structure for capital k and productivity A is summarized below:

Volatility Range (σ)	Capital Grid Size (Nk)	Productivity Grid Size (NA)	Approx. A-Support (±σ)
σ ≤ 0.03	160	9	±3.0σ
0.03 < σ ≤ 0.10	240	13	±3.8σ
0.10 < σ ≤ 0.20	320	17	±4.2σ
0.20 < σ ≤ 0.30	400	21	±4.5σ

 https://doi.org/10.26219/heal.aueb.9687

The capital grid typically spans $k \in [0.2k*, 4k*]$, where $k*$ denotes the deterministic steady-state level, ensuring that both low- and high-capital regimes are represented. This adaptive discretization ensures that global methods such as Value Function Iteration (VFI) and Endogenous Grid Method (EGM) maintain accuracy even under extreme volatility conditions.

7.3 Neural-Network Training Data

For the neural-network solvers, two types of training data are employed in equal proportions (15,000 points each for 30,000 total):

- Correlated economic samples, generated by simulating the economy under the Bellman-optimal (or Endogenous Grid Method (EGM)) policy until it reaches its ergodic distribution. These points represent the most likely states of the model and ensure that the network learns within economically meaningful regions of the state space.
- Uniformly distributed samples, drawn across the full support of k and A grids. They guarantee that the network generalizes beyond the simulated trajectories and remains stable when the system deviates from the steady state.

Combining both datasets in equal proportions allows the Neural Networks to achieve strong generalization while avoiding overfitting to any specific region of the state space.

7.4 Steady-State Normalization

Under the baseline calibration, the deterministic steady-state values are:

$$k^* = (\alpha\beta)^{\frac{1}{1-\alpha}} = 0.166421, \qquad c^* = (k^*)^a - k^* = 0.417511.$$

These benchmarks are used to normalize residuals and impulse-response deviations across methods. All reported errors—Euler, and policy distances—are therefore unit-free and comparable between local and global solvers.

7.5 Conclusion

The calibration described above provides a unified framework for assessing numerical accuracy, stability, and computational efficiency across all solution methods. In the following chapter, each approach, Perturbation, Value Function Iteration (VFI), Endogenous Grid Method (EGM), and the two Neural-Network variants, is evaluated under identical conditions using the residual and distance metrics defined in Chapter 6.

https://doi.org/10.26219/heal.aueb.9687

Chapter 8. Results

This chapter presents the empirical findings from the stress test of the real business cycle model under increasing levels of productivity volatility, σ. The analysis evaluates five solution methods—first- and second-order Perturbation (Pert-1 and Pert-2), Value Function Iteration (VFI) via Bellman-optimal iteration (Bellman), the Endogenous Grid Method (EGM), and two variants of Neural Network approximation: NN (global), which combines distillation from the Endogenous Grid Method (EGM) policy followed by Euler-equation fine-tuning, and NN (equation), which trains solely on the Euler residual from random initialization. All results are obtained under the unified calibration described in Chapter 7, using adaptive state-space grids and Gauss–Hermite quadrature tailored to each volatility level. Performance is assessed across four dimensions: equilibrium accuracy (Euler residuals), policy function approximation error, impulse response function (IRF) fidelity, and computational efficiency.

8.1 Euler Equation Residuals

The Euler equation residual provides a unit-free measure of how closely a solution satisfies the model's intertemporal first-order optimality condition. A lower (more negative) log10 residual indicates higher precision. Table 8.1 summarizes the key statistics (Median, 95th percentile, and Maximum) for each method and volatility level.

Table 8.1: Euler Equation Residuals ($\log_{10}$) Across Methods and Volatility Levels
Lower (more negative) values indicate higher precision.

sigma	method	Euler Median	Euler 95	Euler max
0.005	EGM	-4.668471	-4.5601	-4.52944
0.005	Pert-1	-4.812016	-4.781	-4.77338
0.005	Pert-2	-4.695369	-3.7496	-3.34793
0.005	NN(global)	-4.431343	-4.3357	-3.98305
0.005	NN(equation)	-4.432818	-4.3717	-4.32142
0.01	EGM	-4.70214	-4.5628	-4.52851
0.01	Pert-1	-4.208345	-4.1788	-4.17679
0.01	Pert-2	-4.115546	-3.1413	-2.73645
0.01	NN(global)	-4.163292	-4.1096	-4.05604
0.01	NN(equation)	-4.300995	-3.9867	-3.73873
0.02	EGM	-5.179525	-4.6387	-4.4826
0.02	Pert-1	-3.606095	-3.5767	-3.57585

sigma	method	Euler Median	Euler 95	Euler max
0.02	Pert-2	-3.518384	-2.5405	-2.13214
0.02	NN(global)	-4.553081	-4.4393	-4.15303
0.02	NN(equation)	-4.191204	-4.0329	-3.97188
0.03	EGM	-4.639602	-4.3919	-4.24164
0.03	Pert-1	-3.253364	-3.2242	-3.22384
0.03	Pert-2	-3.153087	-2.1869	-1.78268
0.03	NN(global)	-3.795229	-3.7114	-3.61376
0.03	NN(equation)	-4.359544	-4.1691	-4.10094
0.05	EGM	-4.263587	-4.0444	-3.98905
0.05	Pert-1	-2.809398	-2.7802	-2.7799
0.05	Pert-2	-2.712985	-1.7424	-1.31189
0.05	NN(global)	-4.93527	-4.1155	-3.58298
0.05	NN(equation)	-4.537574	-4.2279	-4.12607
0.07	EGM	-3.851376	-3.6925	-3.63841
0.07	Pert-1	-2.515891	-2.4875	-2.48717
0.07	Pert-2	-2.420617	-1.4443	-1.01368
0.07	NN(global)	-4.318619	-3.8852	-3.67792
0.07	NN(equation)	-3.781339	-3.621	-3.5877
0.1	EGM	-3.487781	-3.3526	-3.33561
0.1	Pert-1	-2.205707	-2.1767	-2.17636
0.1	Pert-2	-2.118556	-1.1403	-0.70201
0.1	NN(global)	-3.930399	-3.6696	-3.15635
0.1	NN(equation)	-4.398745	-3.9062	-3.6408
0.15	EGM	-3.277038	-3.153	-3.13163
0.15	Pert-1	-1.850139	-1.822	-1.82168
0.15	Pert-2	-1.753956	-0.7858	-0.3288
0.15	NN(global)	-4.342343	-3.7735	-3.4235
0.15	NN(equation)	-4.281165	-3.8971	-3.71037
0.2	EGM	-3.012632	-2.8964	-2.8814
0.2	Pert-1	-1.596863	-1.5687	-1.56829
0.2	Pert-2	-1.4957	-0.4869	-0.07099
0.2	NN(global)	-4.01799	-3.6722	-3.16208

 https://doi.org/10.26219/heal.aueb.9687

sigma	method	Euler Median	Euler 95	Euler max
0.2	NN(equation)	-4.006969	-3.7258	-3.18626

The results reveal several key patterns. First, the Endogenous Grid Method (EGM) consistently produces excellent results across all summary statistics, with median residuals around -4.7 and 95th percentiles around -4.6 for low to moderate volatility ($\sigma \leq 0.05$). This confirms Endogenous Grid Method's status as a highly accurate global solver for this class of models. Second, the performance of the Perturbation methods is highly dependent on the volatility level. For very low σ (0.005), both Pert-1 and Pert-2 show strong median accuracy, even surpassing Endogenous Grid Method (EGM). However, the maximum residuals for Pert-2 deteriorate significantly faster than those for Pert-1 as volatility increases. For instance, at σ=0.05, Pert-2's maximum residual is -1.31, compared to -2.78 for Pert-1. This aligns with the theoretical expectation that higher-order approximations can become less stable far from the steady state, a phenomenon highlighted in the stress test of Chapter 5.

Third, and most notably, the Neural Network methods demonstrate robust median performance and reveal an important distinction in their tail behaviour. Contrary to conventional wisdom that distillation from an accurate teacher should uniformly improve performance, NN(equation) consistently demonstrates superior accuracy in the tails of the state space compared to NN(global). This pattern is evident across multiple volatility levels::

- At σ=0.01: NN(equation) has better maximum residual (-3.74 vs -4.06)
- At σ=0.03: NN(equation) shows superior 95th percentile (-4.17 vs -3.71) and maximum (-4.10 vs -3.61)
- At σ=0.1: NN(equation) maintains better 95th percentile (-3.91 vs -3.67) and maximum (-3.64 vs -3.16)

This surprising result suggests that direct minimization of Euler equation residuals enforces more robust generalization to extreme states, whereas distillation-based learning may inherit limitations or approximation biases from the teacher policy in regions underrepresented in training data. The purely equation-based approach appears fundamentally better equipped to capture the full complexity of optimal policies throughout the entire state space. The NN(global) method, despite benefiting from distillation from the accurate Endogenous Grid Method (EGM) policy, appears to inherit some limitation in capturing the full complexity of the optimal policy in regions that are underrepresented in the training data.

https://doi.org/10.26219/heal.aueb.9687

Both Neural Network approaches maintain respectable median Euler residuals, typically between -4.4 and -3.9 across the volatility spectrum, demonstrating their viability as solution methods. However, the superior tail performance of NN(equation) highlights the potential advantage of training directly against the model's fundamental equilibrium conditions rather than relying on imitation learning from another numerical method.

In summary, while all non- Perturbation methods maintain good median Euler accuracy, Endogenous Grid Method (EGM) sets the standard for overall reliability. The Perturbation methods are accurate locally but exhibit significant errors in the tails of the state space as volatility rises, validating the findings of the local validity stress test. The Neural Network approaches demonstrate that equation-based learning can achieve better tail performance than distillation-based approaches.

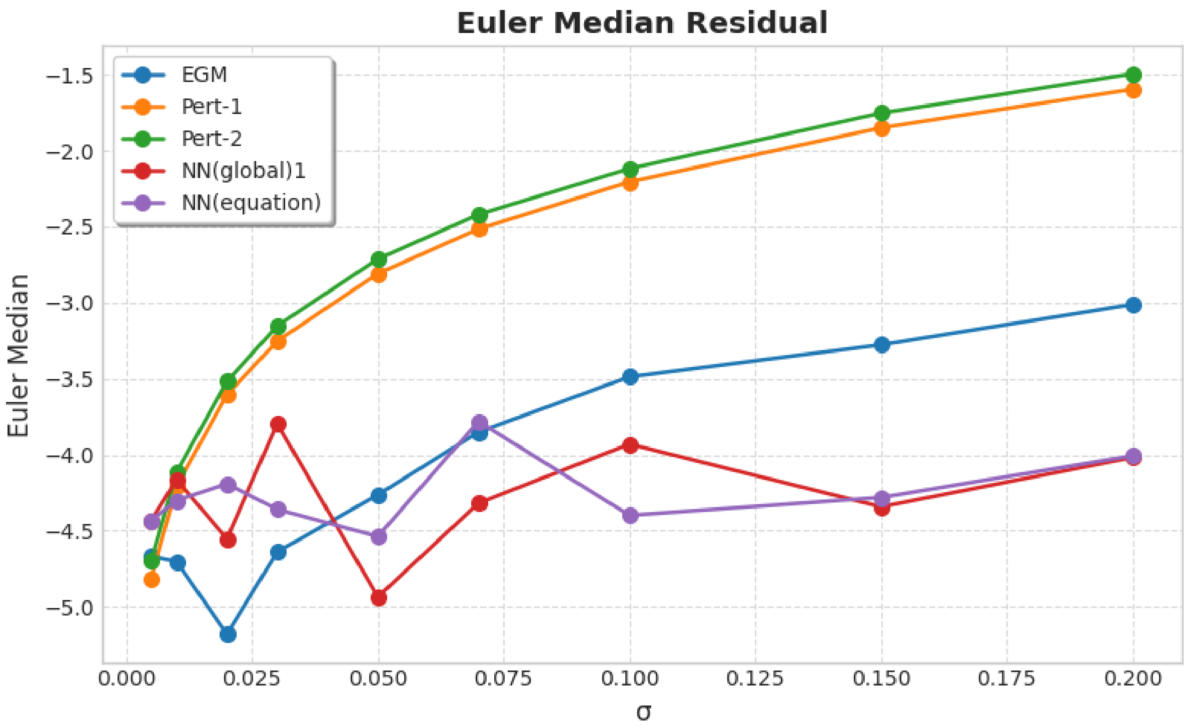


Figure 8.1: Euler Residual Medians Across Volatility Spectrum
Median Euler equation residuals ($\log_{10}$) for all solution methods. EGM and Neural Networks maintain strong median performance, while Perturbation methods show systematic deterioration with increasing volatility.

https://doi.org/10.26219/heal.aueb.9687

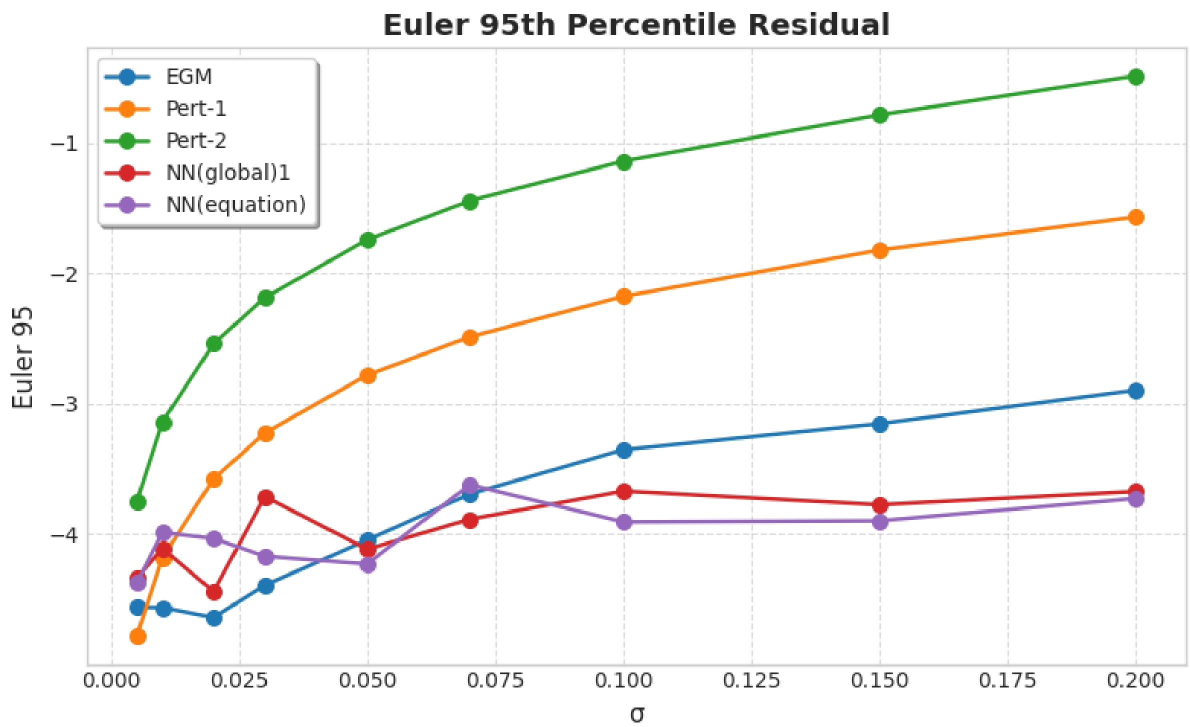


Figure 8.2: Euler Residual 95th Percentiles Across Volatility Spectrum

95th percentile of Euler equation residuals ($\log_{10}$), indicating tail performance. NN(equation) consistently outperforms NN(global) in the tails, particularly at moderate to high volatility levels.

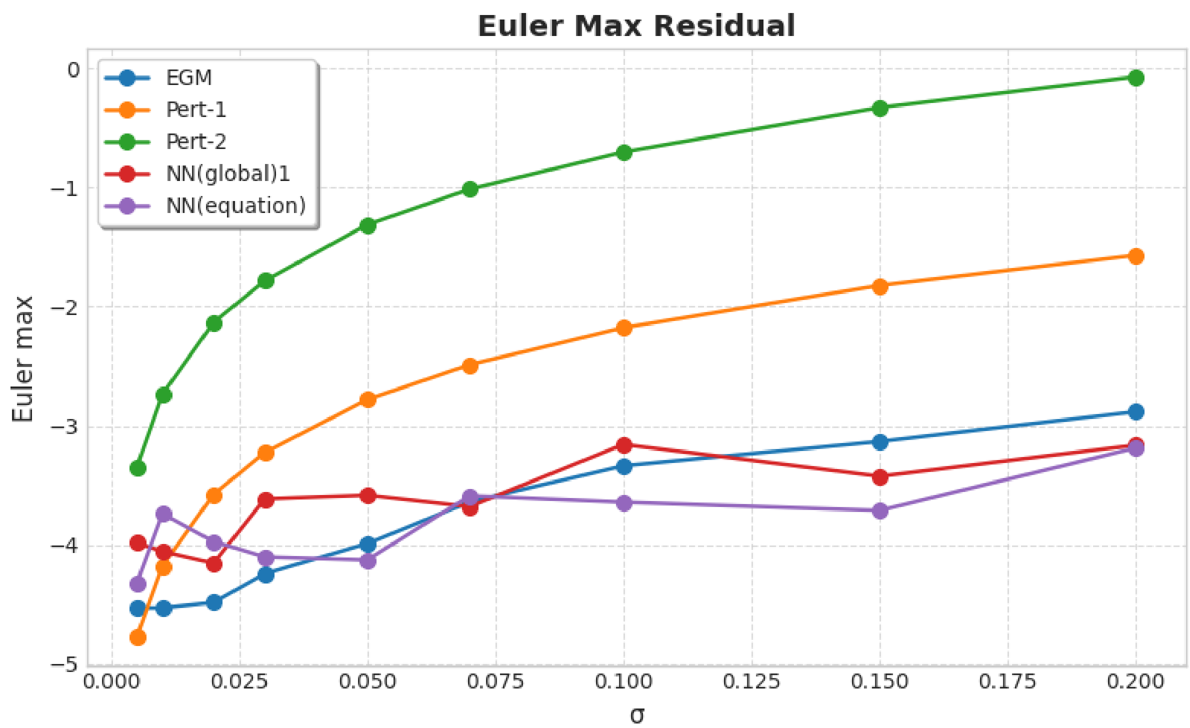


Figure 8.3: Maximum Euler Residuals Across Volatility Spectrum

Worst-case Euler equation errors ($\log_{10}$). The superior tail robustness of NN(equation) is most evident here, with significantly better maximum residuals than NN(global) across most volatility levels.

https://doi.org/10.26219/heal.aueb.9687

8.2 Policy Function Errors

The accuracy of the policy functions for next-period capital, k′, is measured by the Root Mean Square Error (RMSE) and the sup-norm error relative to the Value Function Iteration (VFI) benchmark. These metrics, shown in Table 8.2, evaluate the average and worst-case deviations, respectively.

Table 8.2: Policy Function Errors Relative to VFI Benchmark

Root Mean Square Error (RMSE) and sup-norm errors for next-period capital policy.

sigma	method	Pol_RMSE	Pol_sup
0.005	EGM	0.000458	0.000843
0.005	Pert-1	0.000484	0.00112
0.005	Pert-2	0.000559	0.001669
0.005	NN(global)	0.000458	0.000894
0.005	NN(equation)	0.000458	0.000861
0.01	EGM	0.000475	0.000897
0.01	Pert-1	0.000491	0.001152
0.01	Pert-2	0.000812	0.002779
0.01	NN(global)	0.000475	0.000897
0.01	NN(equation)	0.000473	0.000884
0.02	EGM	0.000455	0.000844
0.02	Pert-1	0.000485	0.001288
0.02	Pert-2	0.001422	0.005241
0.02	NN(global)	0.000455	0.000848
0.02	NN(equation)	0.000455	0.000856
0.03	EGM	0.000454	0.000853
0.03	Pert-1	0.000486	0.001341
0.03	Pert-2	0.002182	0.008307
0.03	NN(global)	0.000455	0.000892
0.03	NN(equation)	0.000453	0.000852
0.05	EGM	0.000547	0.001137
0.05	Pert-1	0.000894	0.003915
0.05	Pert-2	0.009114	0.036938
0.05	NN(global)	0.000547	0.001105
0.07	EGM	0.000541	0.001093
0.07	Pert-1	0.001044	0.00494
0.07	Pert-2	0.014558	0.061046
0.07	NN(global)	0.000542	0.001096
0.07	NN(equation)	0.000541	0.001073
0.1	EGM	0.000549	0.001173
0.1	Pert-1	0.001478	0.007234
0.1	Pert-2	0.025655	0.108364
0.1	NN(global)	0.000554	0.001177
0.1	NN(equation)	0.000553	0.001178
0.15	EGM	0.000499	0.001148
0.15	Pert-1	0.008507	0.095069
0.15	Pert-2	0.068768	0.243022
0.15	NN(global)	0.005714	0.073483
0.15	NN(equation)	0.005749	0.07423
0.2	EGM	0.000485	0.001281
0.2	Pert-1	0.047541	0.326075
0.2	Pert-2	0.098392	0.260438
0.2	NN(global)	0.041099	0.286944
0.2	NN(equation)	0.041132	0.28693

https://doi.org/10.26219/heal.aueb.9687

sigma	method	Pol_RMSE	Pol_sup
0.05	NN(equation)	0.000547	0.001139

The policy error analysis reinforces the conclusions from the Euler residuals. Endogenous Grid Method (EGM) and the Neural Network methods produce policy functions that are virtually indistinguishable from the Value Function Iteration (VFI) benchmark at low volatility, with RMSEs on the order of 10^{-4}. The NN(global) and NN(equation) policies are remarkably precise, with their RMSE values often matching Endogenous Grid Methods's exactly, demonstrating their capacity for high-fidelity approximation.

The Perturbation methods show a clear increase in policy error with volatility. While Pert-1 and Pert-2 start with errors similar to the global methods at σ=0.005, their sup-norm errors grow substantially. For example, by σ=0.1, Pert-2's sup-norm error is 0.0257, and by σ=0.2, it reaches 0.0984, indicating regions of the state space where the local approximation fails catastrophically. Pert-1 remains more stable but still shows a marked increase in error.

The latter confirms that for local dynamics and small shocks, Perturbation -derived policies are adequate. However, for studying large deviations or high-volatility regimes, the global methods—Endogenous Grid Method (EGM) and Neural Networks—are necessary to ensure policy accuracy across the entire state space.

https://doi.org/10.26219/heal.aueb.9687

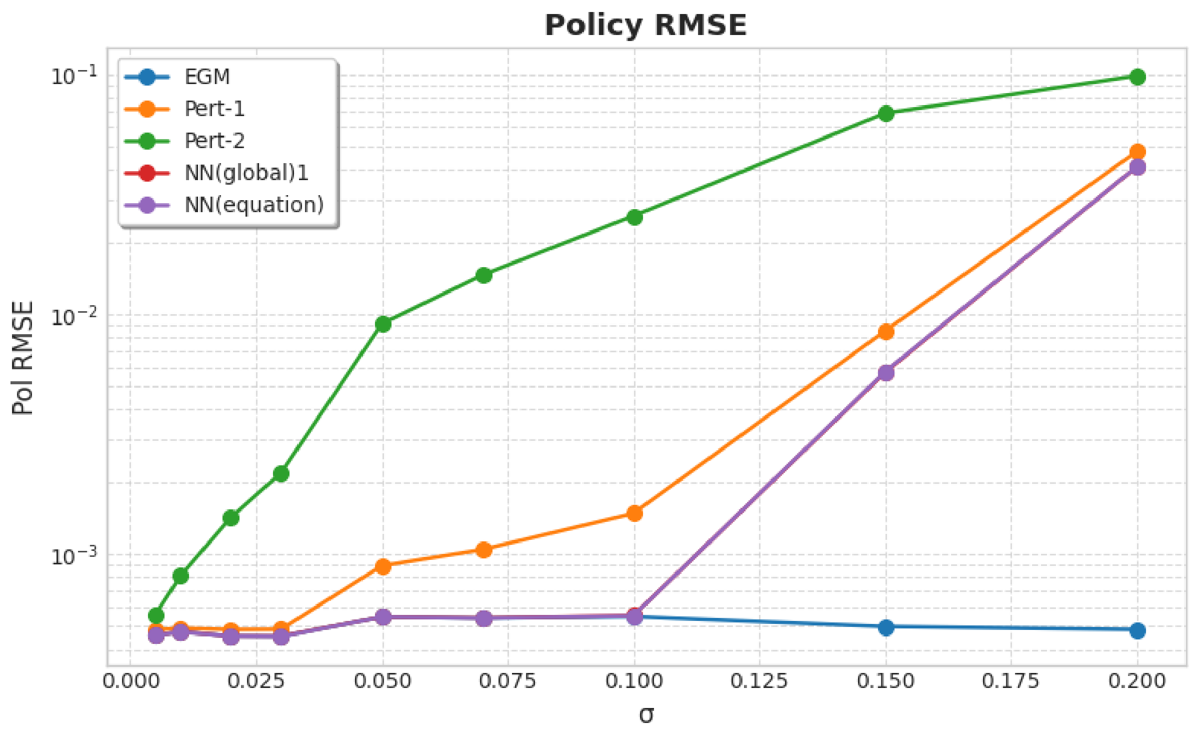


Figure 8.4: Policy Function RMSE Across Volatility Levels

Root Mean Square Error of next-period capital policies relative to VFI benchmark. EGM and Neural Networks maintain consistently low errors, while Perturbation methods show volatility-dependent deterioration.

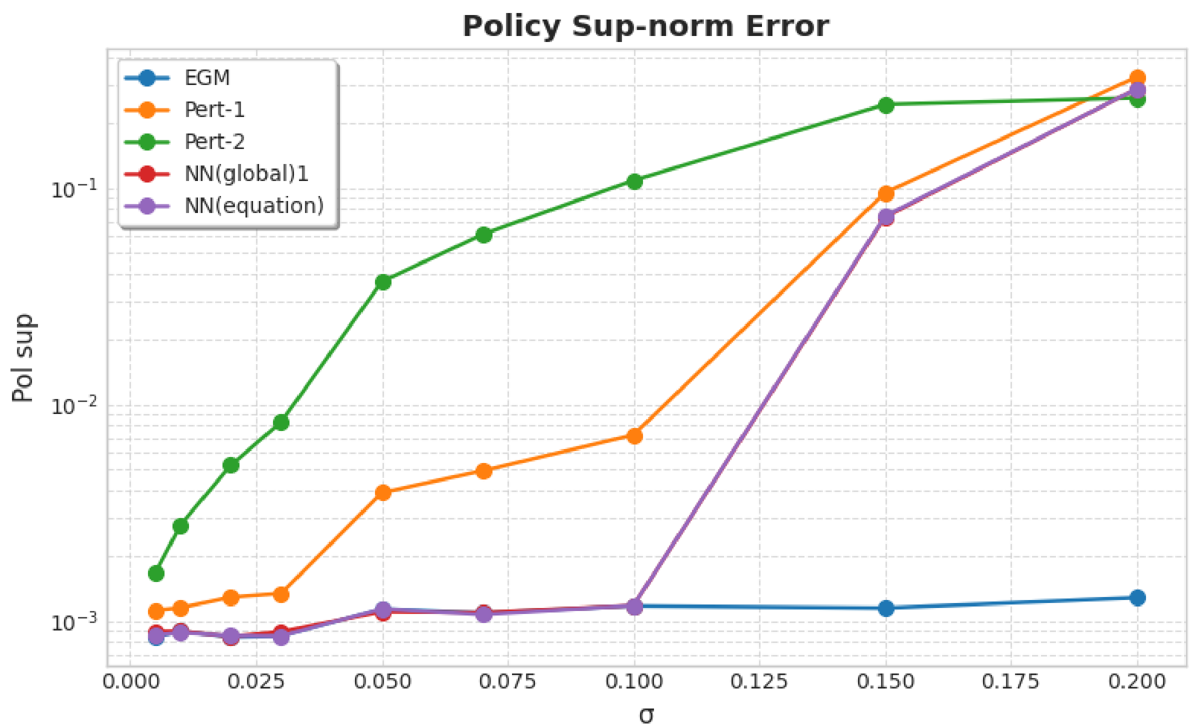


Figure 8.5: Policy Function Sup-Norm Errors Across Volatility Levels

Worst-case policy deviations from VFI benchmark. The global accuracy advantage of EGM and Neural Networks is particularly evident in this maximum error metric.

https://doi.org/10.26219/heal.aueb.9687

8.3 Impulse Response Function (IRF) Deviations

To assess dynamic accuracy, the maximum absolute deviations of the capital (IRF_k_sup) and consumption (IRF_c_sup) impulse response functions from the benchmark are recorded. These values, presented as percentages, are shown in Table 8.3.

Table 8.3: Impulse Response Function Maximum Deviations (%)
Sup-norm errors for capital (IRF_k_sup) and consumption (IRF_c_sup) impulse responses relative to VFI benchmark.

sigma	method	IRF_k_sup	IRF_c_sup	sigma	method	IRF_k_sup	IRF_c_sup
0.005	EGM	0.000407	0.000481	0.07	EGM	0.001205	0.000597
0.005	Pert-1	0.000409	0.000481	0.07	Pert-1	0.00195	0.00075
0.005	Pert-2	0.000407	0.000484	0.07	Pert-2	0.00195	0.000742
0.005	NN(global)	0.000405	0.000479	0.07	NN(global)	0.001214	0.000591
0.005	NN(equation)	0.000405	0.00048	0.07	NN(equation)	0.00117	0.000581
0.01	EGM	0.000463	0.000258	0.1	EGM	0.000601	0.000601
0.01	Pert-1	0.000455	0.000259	0.1	Pert-1	0.00195	0.000641
0.01	Pert-2	0.00045	0.000257	0.1	Pert-2	0.00195	0.000633
0.01	NN(global)	0.000455	0.000259	0.1	NN(global)	0.000549	0.000549
0.01	NN(equation)	0.000479	0.000267	0.1	NN(equation)	0.000569	0.000569
0.02	EGM	0.000506	0.000288	0.15	EGM	0.000608	0.000252
0.02	Pert-1	0.000459	0.000259	0.15	Pert-1	0.004043	0.002294
0.02	Pert-2	0.000454	0.000257	0.15	Pert-2	0.004043	0.002307
0.02	NN(global)	0.000515	0.000292	0.15	NN(global)	0.000629	0.000294
0.02	NN(equation)	0.000522	0.000297	0.15	NN(equation)	0.000657	0.00031
0.03	EGM	0.000579	0.000344	0.2	EGM	0.000474	0.000474
0.03	Pert-1	0.000467	0.000261	0.2	Pert-1	0.006447	0.004105
0.03	Pert-2	0.000462	0.000257	0.2	Pert-2	0.006447	0.004118
0.03	NN(global)	0.000548	0.000318	0.2	NN(global)	0.000568	0.000568
0.03	NN(equation)	0.000587	0.000346	0.2	NN(equation)	0.000573	0.000573
0.05	EGM	0.000753	0.000594				
0.05	Pert-1	0.000439	0.000519				
0.05	Pert-2	0.000433	0.000527				
0.05	NN(global)	0.000749	0.000601				
0.05	NN(equation)	0.000749	0.000601				

https://doi.org/10.26219/heal.aueb.9687

The Impulse Response Function (IRF) errors are consistently very small for all methods at low volatility, confirming that local dynamics around the steady state are well-captured even by Perturbation. However, as volatility increases, the performance diverges.

The Endogenous Grid Method (EGM) and Neural Network methods maintain low Impulse Response Function errors throughout the volatility range. For capital (IRF_k_sup), their errors remain below 0.07% for $\sigma \leq 0.1$, only rising to around 0.06% for the NNs at σ=0.2. This demonstrates their robustness in reproducing the model's transition dynamics.

In contrast, the Perturbation methods show a different pattern. Their Impulse Response Function (IRF) errors for consumption (IRF_c_sup) initially decrease as volatility rises from 0.005 to 0.01 but then increase non-monotonically. For example, Pert-1's IRF_c_sup rises to 0.641% at σ=0.1. This non-monotonicity is consistent with the "hump-shaped" pattern of reliability identified in Chapter 5, where Perturbation accuracy is maximized at moderate volatility before deteriorating.

Overall, the Impulse Response Function analysis confirms that global methods provide reliable dynamic paths under all volatility regimes, whereas the accuracy of Perturbation methods is conditional on the shock environment.

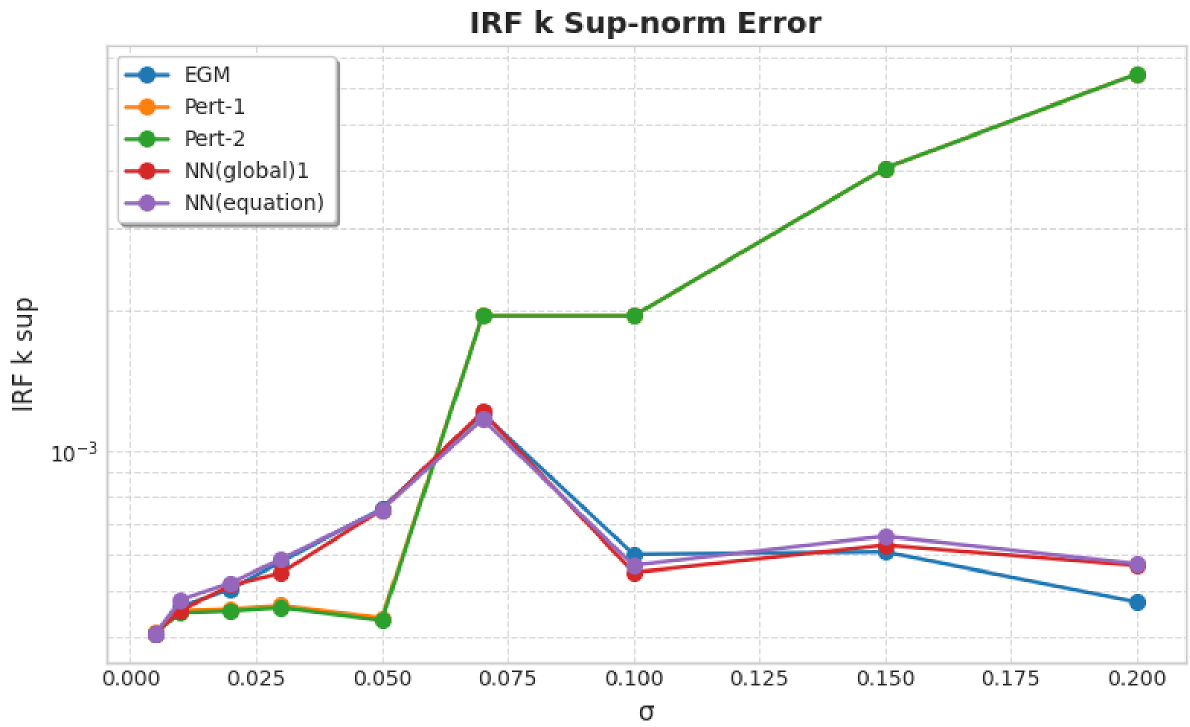


Figure 8.6: Capital IRF Maximum Deviations Across Volatility

Maximum percentage deviations in capital impulse responses from VFI benchmark. The reliability of global methods

https://doi.org/10.26219/heal.aueb.9687

contrasts with the conditional accuracy of Perturbation approaches.

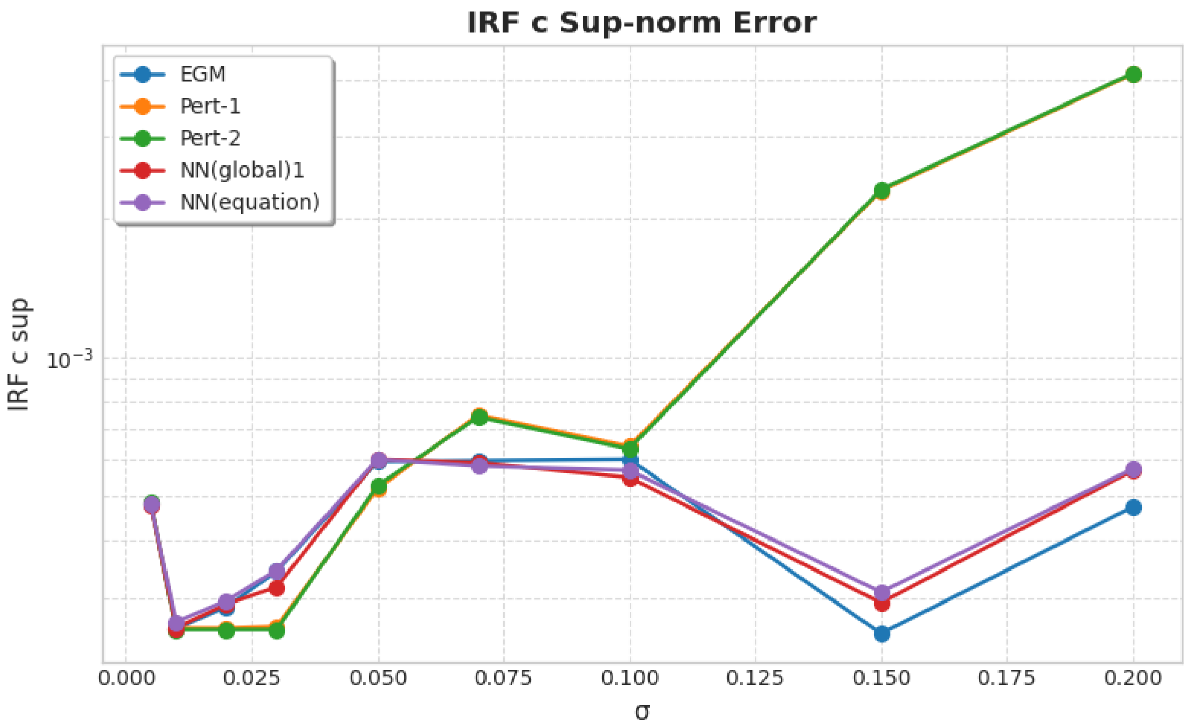
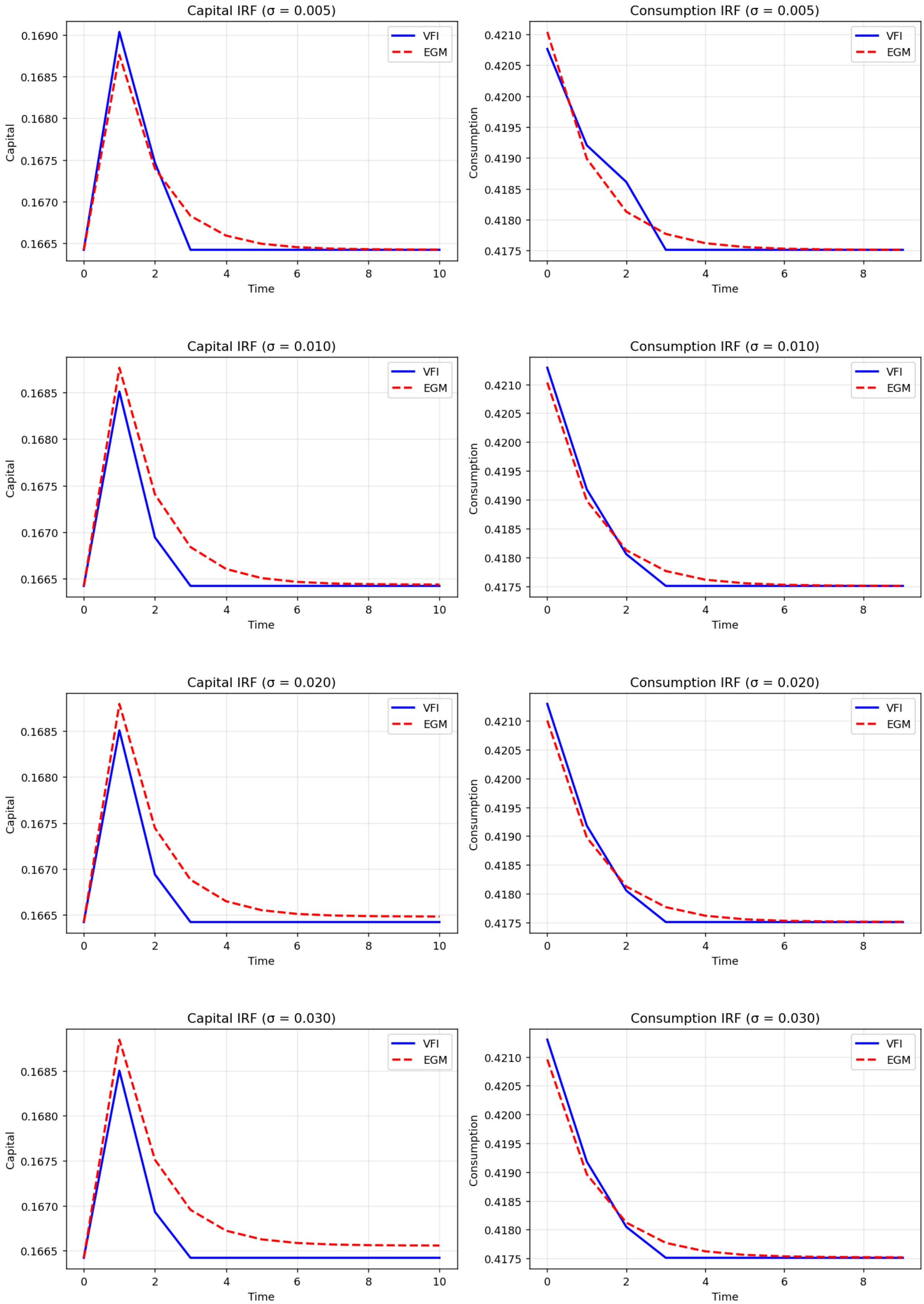


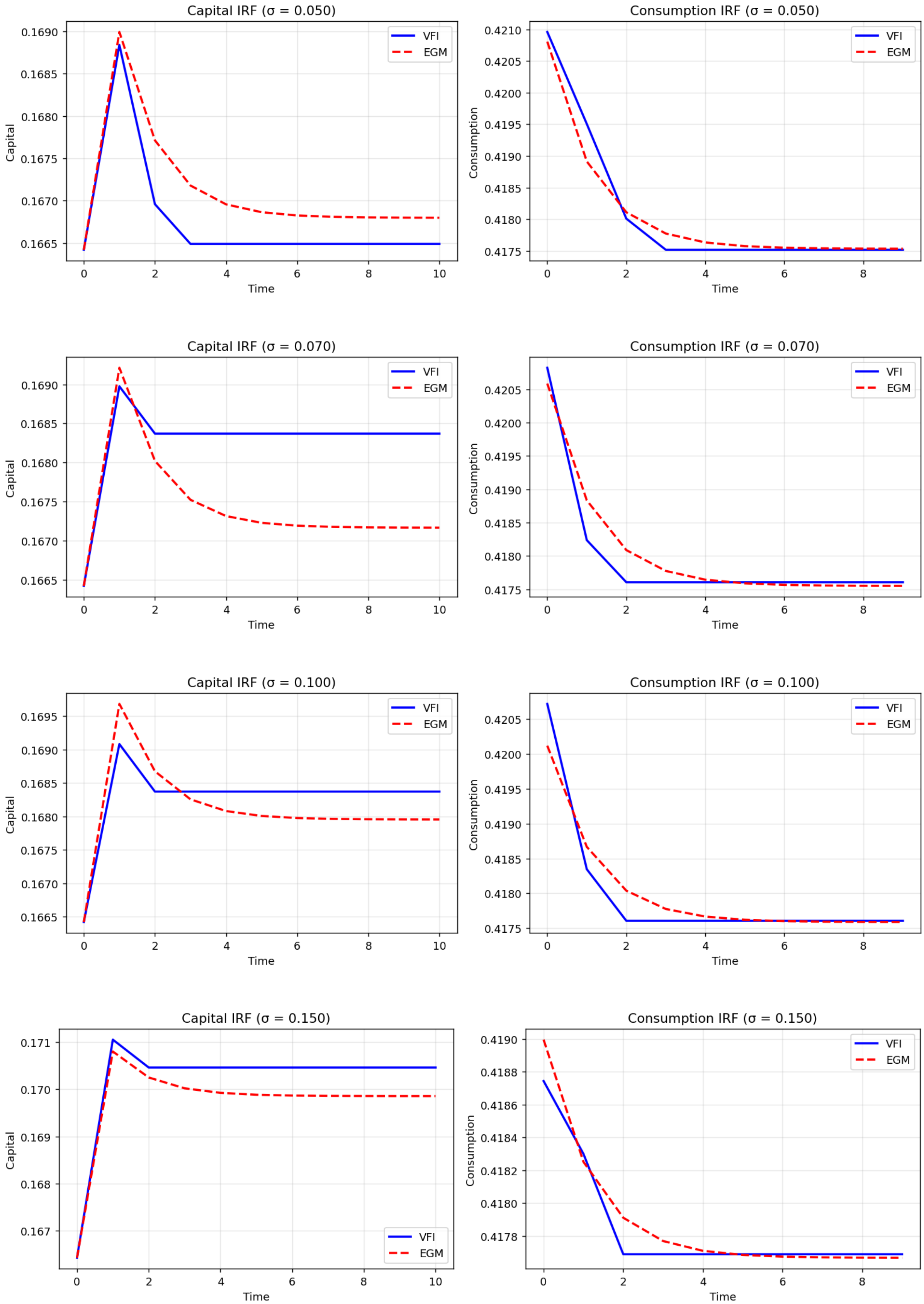
Figure 8.7: Consumption IRF Maximum Deviations Across Volatility

Maximum percentage deviations in consumption impulse responses. The non-monotonic pattern for Perturbation methods reflects the hump-shaped reliability boundary identified in Chapter 5.

To visually demonstrate these findings, Figures 8.8, 8.9 and 8.10 present the impulse response dynamics directly. Figure 8.8 compares the VFI and EGM solutions for a 1% productivity shock across nine representative volatility levels. The near-perfect alignment of the paths confirms that Endogenous Grid Method (EGM) replicates the Value Function Iteration (VFI) benchmark dynamics with exceptional fidelity, validating its use as a reliable global solver. Figures 8.9 and 8.10 illustrates how the model's transition dynamics evolve with increasing volatility using the VFI benchmark. As σ rises, the responses become more pronounced and persistent, with capital accumulation showing longer adjustment periods and consumption exhibiting stronger precautionary dynamics, effects that local Perturbation methods struggle to capture accurately.

https://doi.org/10.26219/heal.aueb.9687




Capital IRF (σ = 0.050)
Consumption IRF (σ = 0.050)
Capital IRF (σ = 0.070)
Consumption IRF (σ = 0.070)
Capital IRF (σ = 0.100)
Consumption IRF (σ = 0.100)
Capital IRF (σ = 0.150)
Consumption IRF (σ = 0.150)
Capital
Consumption
Time
VFI
EGM

https://doi.org/10.26219/heal.aueb.9687

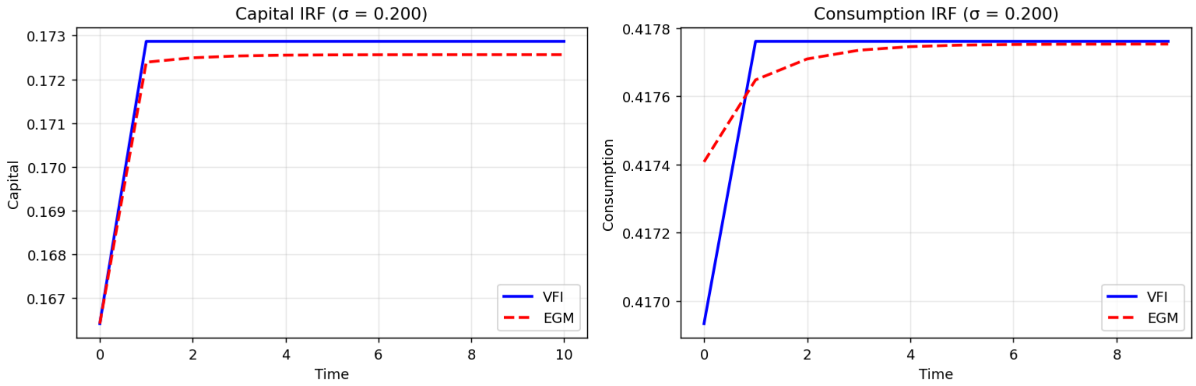


Figure 8.8: VFI vs EGM Impulse Response Comparison for 1% Productivity Shock

Impulse response functions for capital (left) and consumption (right) following a one-time 1% productivity shock. Both Value Function Iteration (VFI) and Endogenous Grid Method (EGM) solutions are shown for all nine volatility levels (σ = 0.005-0.2). The near-perfect overlap demonstrates EGM's accuracy as a global benchmark across different volatility regimes.

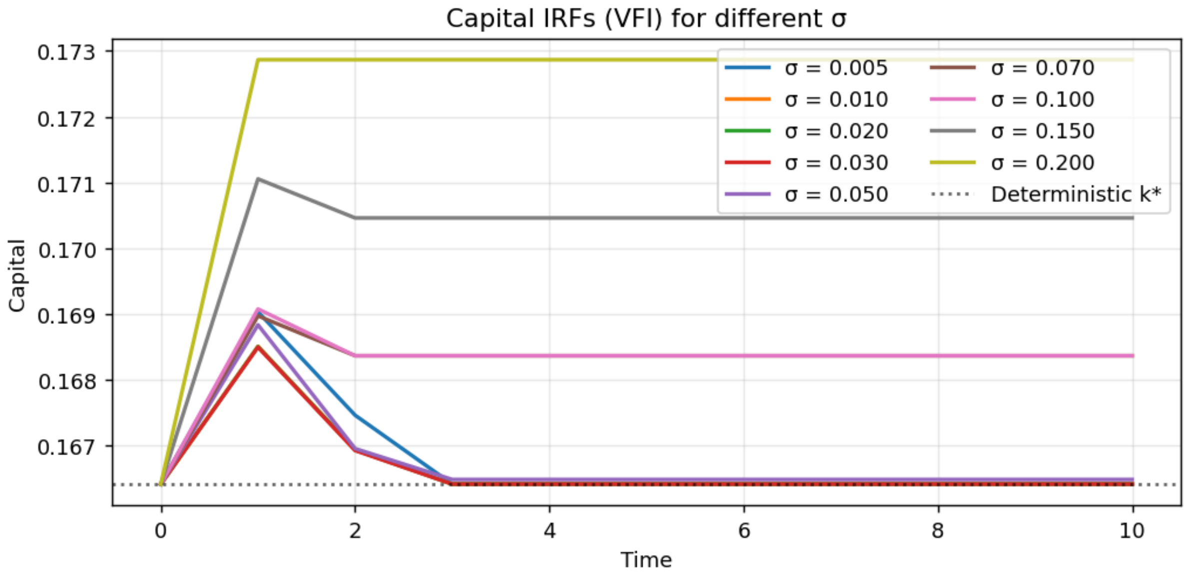


Figure 8.9: VFI Impulse Response Dynamics Across Volatility Levels

Benchmark impulse responses from Value Function Iteration (VFI) for capital across the full volatility spectrum (σ = 0.005-0.2). Higher volatility amplifies and prolongs the economic response, with capital showing increased persistence during the transition.

https://doi.org/10.26219/heal.aueb.9687

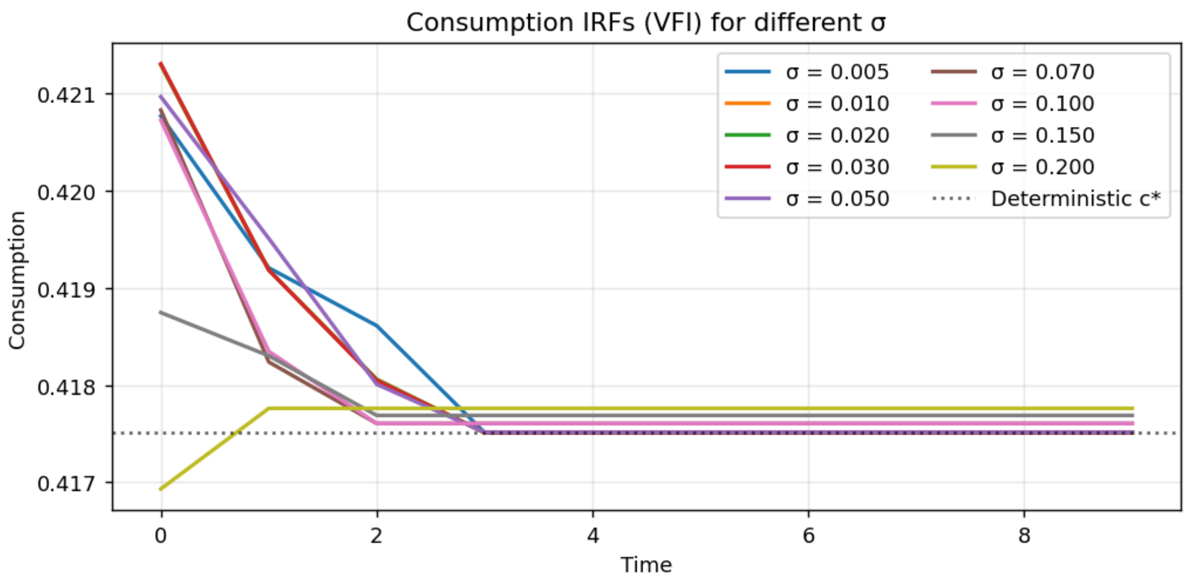


Figure 8.10: VFI Impulse Response Dynamics Across Volatility Levels

Benchmark impulse responses from Value Function Iteration (VFI) for capital across the full volatility spectrum (σ = 0.005-0.2). Higher volatility amplifies and prolongs the economic response, with consumption exhibiting stronger precautionary behavior during the transition.

8.4 Computational Efficiency

The computational cost of each method is a critical practical consideration. Table 8.4 reports the total solution time in seconds for the Value Function Iteration (VFI), Endogenous Grid Method (EGM), and Neural Network (NN) methods, while Table 8.5 details the Neural Network forward pass time per evaluation.

Table 8.4: Total Solution Time by Method (Seconds)

sigma	VFI	EGM	NN(global)	NN(equation)
0.005	2.17	0.78	1133.79	1910.58
0.01	0.96	0.29	1146.31	1904.24
0.02	0.96	0.28	1143.75	1889.9
0.03	0.95	0.28	1139.71	1892.95
0.05	2.94	1.1	1512.56	2516.01
0.07	3.29	0.69	1507.23	2550.83
0.1	2.67	0.74	1523.03	2563.35
0.15	6.37	1.31	1767.26	3003.44
0.2	17.45	2.65	1495.02	2564.69

 https://doi.org/10.26219/heal.aueb.9687

A critical observation not shown in this table is the computational cost of the Perturbation methods. The first- and second-order Perturbation solutions were obtained in 0.64 and 2.06 seconds, respectively, with these times remaining constant across all volatility levels (σ).

Table 8.5: Neural Network Forward Pass Time per Evaluation (Microseconds)
Policy evaluation speed post-training.

sigma	NN(Global)	NN(equation)
0.005	249.18 μs	188.42 μs
0.01	196.84 μs	262.61 μs
0.02	172.56 μs	170.26 μs
0.03	280.54 μs	243.37 μs
0.05	193.37 μs	209.76 μs
0.07	188.71 μs	171.80 μs
0.1	307.72 μs	170.83 μs
0.15	176.22 μs	172.94 μs
0.2	169.36 μs	251.44 μs

The results reveal a stark trade-off between upfront training cost and evaluation speed. The Perturbation methods are not only the fastest for obtaining a single solution but also uniquely efficient as their computation time is invariant to the level of volatility. This is because their algorithm involves a fixed set of matrix operations around the steady state, independent of the shock size. The Endogenous Grid Method (EGM) is the undisputed efficiency champion, solving the model to a high degree of accuracy in a fraction of a second, and is consistently an order of magnitude faster than the Value Function Iteration (VFI) method. This underscores Endogenous Grid Method's computational elegance and its suitability for applications requiring repeated model solutions, such as estimation.

The Neural Network methods have a very high upfront computational cost, with training times ranging from approximately 1100 to 3000 seconds (18-50 min) for NN(global) and NN(equation), respectively. This cost is due to the extensive sampling and iterative gradient descent required during training. However, this investment yields a transformative capability: an exceptionally rapid evaluation speed. Once trained, the policy function can be evaluated via a forward pass in only ~170-300 microseconds. This creates a distinct computational paradigm.

https://doi.org/10.26219/heal.aueb.9687

While Endogenous Grid Method (EGM) is faster for finding the solution, the trained Neural Network is vastly faster at evaluating the solution.

This performance profile defines the ideal use case for the Neural Network approach developed in this thesis. It is specifically superior for research tasks that, for a fixed model calibration, require a massive number of policy evaluations. Examples include:

- Running extensive Monte Carlo simulations with a very large number of stochastic paths to analyse tail risks or convergence properties.
- Conducting detailed welfare analysis that involves integrating value functions over the entire ergodic distribution of the model.
- Performing sensitivity analysis that requires evaluating the model at a dense grid of state points far beyond the resolution of a typical EGM grid.
- Embedding the solved model within a larger computational framework where the policy function needs to be called millions of times.

In these scenarios, the high upfront training cost of the Neural Network is amortized over millions of near-instantaneous evaluations, leading to a significant net reduction in total computational time compared to repeatedly interpolating from a pre-computed EGM grid.

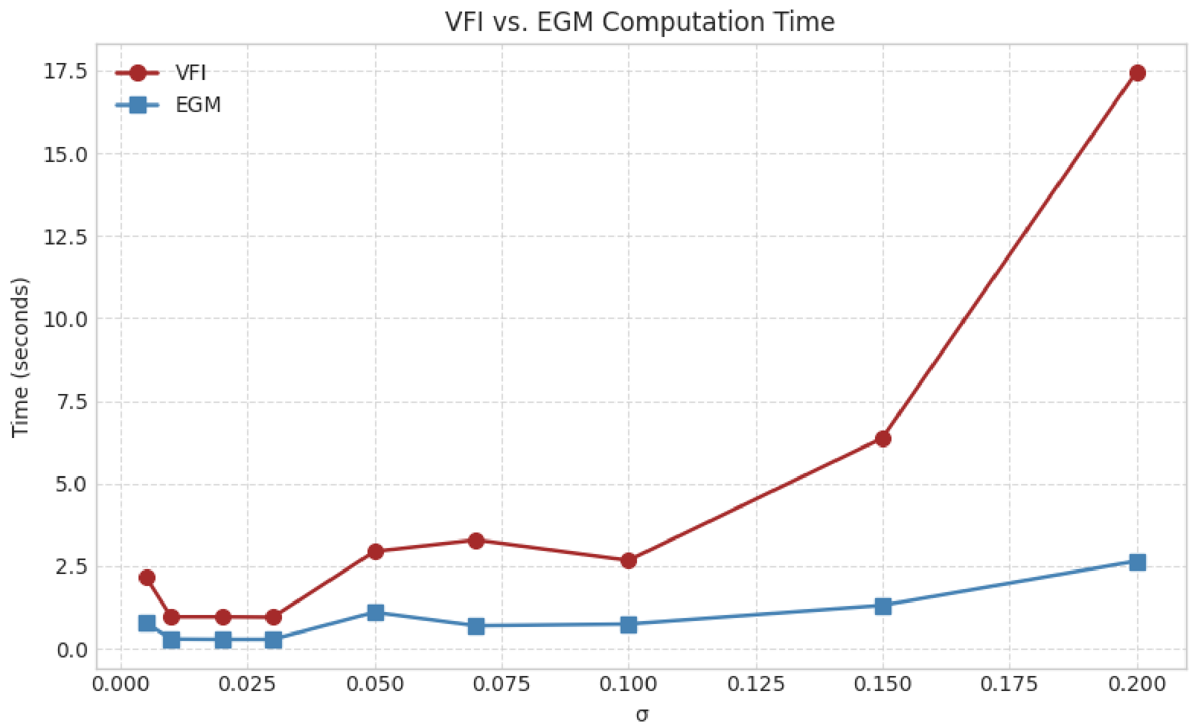


Figure 8.11: Total Solution Time Comparison

Wall-clock time for solution convergence. EGM's efficiency advantage is consistent across volatility levels, while Neural

https://doi.org/10.26219/heal.aueb.9687

Network training represents a fixed computational investment.

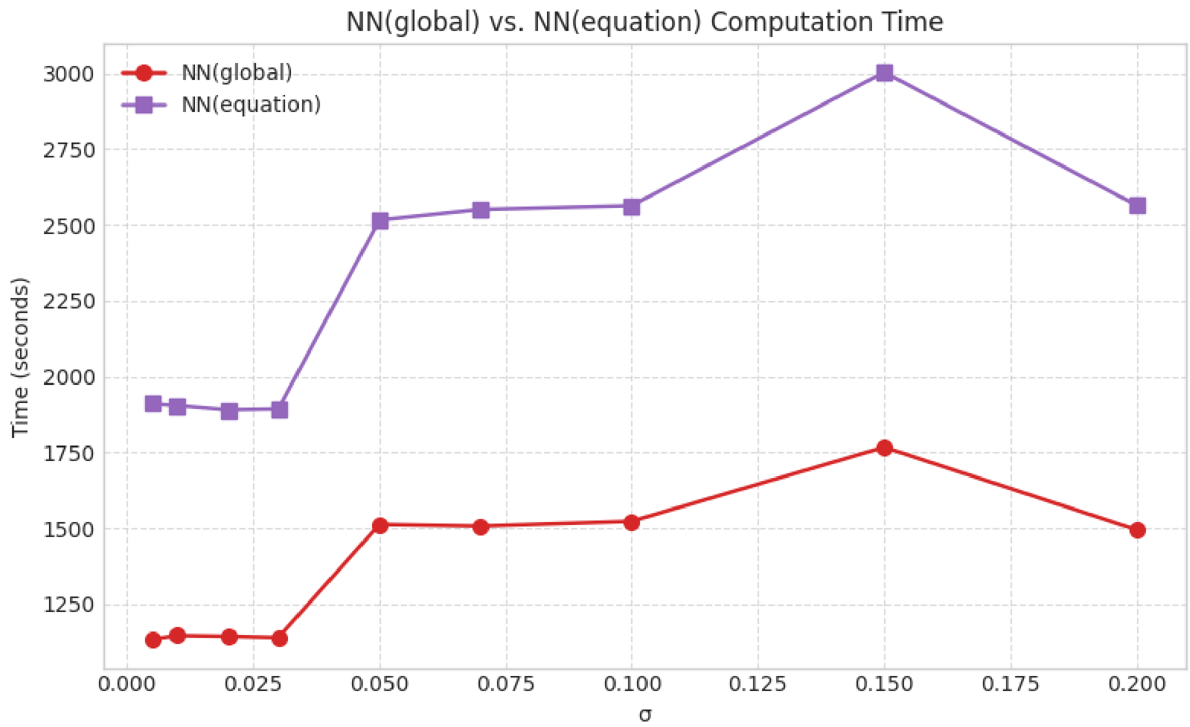


Figure 8.12: Neural Network Training Time Breakdown

Comparison of training duration for NN(global) and NN(equation) variants. The equation-based approach requires longer training but delivers superior tail performance.

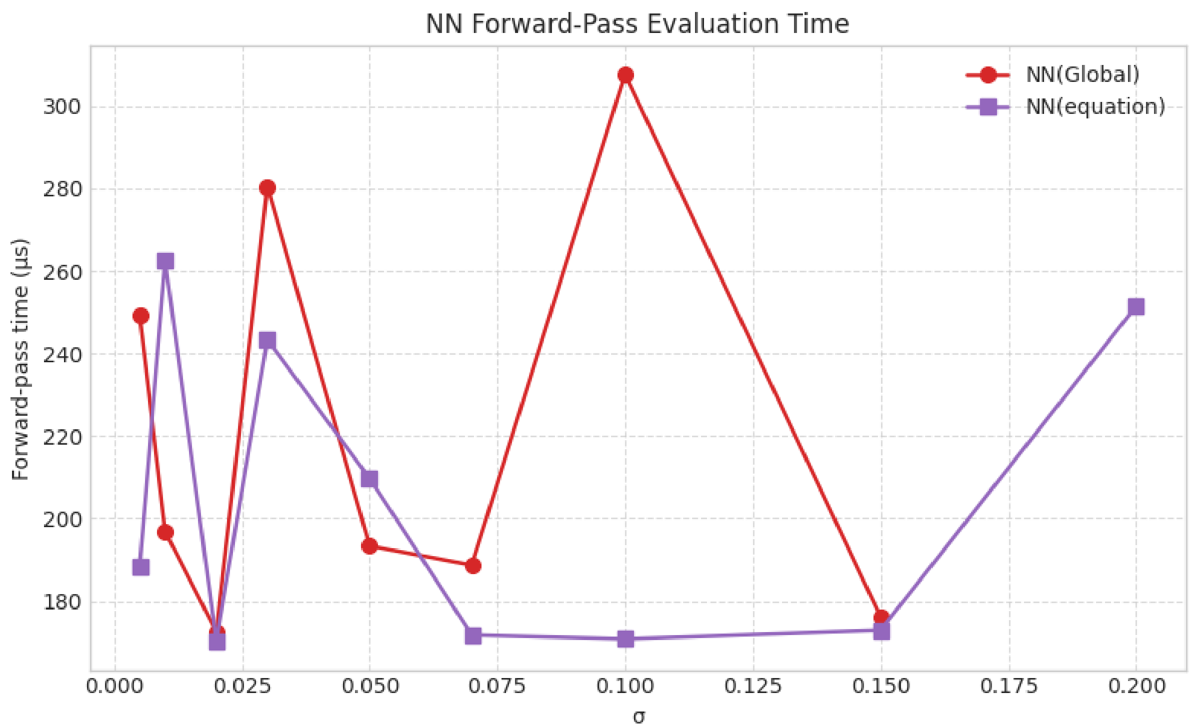


Figure 8.13: Neural Network Evaluation Speed

Forward pass time per policy evaluation in microseconds. Once trained, both Neural Network variants enable near-instantaneous policy evaluation, enabling massive-scale simulation.

https://doi.org/10.26219/heal.aueb.9687

In conclusion, the Neural Network solvers implement a "solve-once, simulate-instantly" workflow. For the specific class of problems involving extremely large-scale simulation and evaluation under a fixed set of model parameters, the Neural Network approach offers a compelling computational advantage, despite its substantial initial time investment. This makes it a powerful tool for high-precision analysis that would be computationally prohibitive with traditional methods.

8.5 Conclusion

This chapter has presented a comprehensive empirical evaluation of five numerical methods for solving a stochastic Real Business Cycle model across a spectrum of productivity volatilities. The results reveal a clear hierarchy of methods, each with distinct strengths and optimal domains of application, thereby providing clear guidance for computational economists.

First, the central trade-off between local efficiency and global reliability is unequivocally demonstrated. Perturbation methods, both first- and second-order, provide a fast and accurate solution for local dynamics around the steady state under low volatility. However, their performance deteriorates markedly as volatility increases, with Euler residuals and policy errors rising by several orders of magnitude. This validates the theoretical limitations of local approximations and quantifies the boundary of their validity established in Chapter 5. For analyses involving large shocks or high-volatility regimes, Perturbation methods are insufficient, and a global solver is required.

Second, among global solvers, the Endogenous Grid Method (EGM) emerges as the benchmark for accuracy and efficiency. It consistently delivers near-perfect Euler residuals, minimal policy errors, and faithful impulse response dynamics across all tested volatility levels. Its computational cost, at just a fraction of a second to a few seconds per solution, makes it the unequivocal choice for most practical applications, including calibration and single-model analysis. Endogenous Grid Method (EGM) sets the standard against which emerging methods must be measured.

Third, the Neural Network approaches represent a powerful and viable new class of global solvers. Both NN(global) and NN(equation) achieve a high degree of accuracy, with policy functions that are often indistinguishable from the EGM and Value Function Iteration (VFI) benchmarks. A particularly significant finding is that the purely equation-based NN(equation) generally outperforms the distillation-based NN(global) in the tails of the state space,

 https://doi.org/10.26219/heal.aueb.9687

demonstrating the advantage of learning directly from the model's fundamental equilibrium conditions. This suggests that for ultimate robustness, the equation-based approach is preferable, despite its slightly longer training time.

Finally, the computational analysis defines the unique niche for Neural Network solvers. While their upfront training cost is substantial, their ability to evaluate the policy function in microseconds unlocks a "solve-once, simulate-instantly" paradigm. This makes them uniquely suited for computational tasks that require a massive number of policy evaluations under a fixed calibration, such as large-scale Monte Carlo studies, detailed welfare analysis over the ergodic distribution, or embedding within larger-scale models.

In summary, the choice of solver is not one-size-fits-all but should be guided by the research question:

- For local analysis and fast calibration, Perturbation methods are adequate.
- For most global solution needs, the Endogenous Grid Method (EGM) offers an ideal balance of speed, accuracy, and reliability.
- For large-scale simulation and evaluation projects, Neural Networks, particularly the equation-based variant, provide a formidable computational advantage after an initial training investment.

This multi-method comparison not only benchmarks the performance of established techniques but also firmly establishes Neural Networks as a credible and powerful tool in the computational economist's toolkit, capable of combining global accuracy with unparalleled evaluation speed for targeted applications.

Chapter 9. Future Research Directions

The empirical results in this thesis demonstrate that Neural Network (NN)-based solvers are not a theoretical framework that waits to be developed, but a new ready to use competitive alternative for solving dynamic stochastic economic models. The strong performance of both NN(Global) and NN(equation) in approximating the policy function of a canonical Real Business Cycle (RBC) model, achieving near-benchmark accuracy while enabling instantaneous evaluation, validates the core premise of this investigation. However, the one sector real business cycle model with i.i.d. shocks represent the simplest meaningful laboratory for such methods. This success opens the door to a richer and more essential research that will

https://doi.org/10.26219/heal.aueb.9687

explore the scalability, robustness and ultimate potential of the Neural Network based techniques.

9.1 Escalating Model Complexity

The promising results on the baseline model establish a foundation, but the true test of Neural Network solvers lies in their ability to handle more economically relevant and computationally demanding settings. A systematic research pathway should progress through stages of increasing complexity, explicitly designed to probe the methods' scalability and their capacity to overcome the limitations of traditional techniques.

1. The Current Model with More Challenging Parameters: The first logical step is to stress-test the existing Neural Network framework within the one-agent model by moving away from the simplifying assumptions of $\delta=1$ and $\rho=0$. Introducing partial depreciation (e.g., $\delta=0.95$) and persistent shocks (e.g., $\rho=0.6$, 0.9) creates a more realistic and computationally intensive environment. This would test the NN's ability to handle a non-triangular state transition matrix and a more complex ergodic distribution without an increase in the nominal number of state variables. This updated framework described above, would be more challenging for the traditional methods that where used throughout this thesis and would favour Neural Network solvers, because of the more nonlinear environment that would occur.
2. Models with Richer Microfoundations: The next stage involves increasing the number of endogenous state and control variables. This includes moving to:
    a. A multi-agent framework: Starting with a simple two-agent model with heterogeneous preferences or wealth, which immediately introduces a second endogenous state variable (the wealth distribution) and tests the NN's ability to approximate policies in a higher-dimensional space.
    b. Endogenous labour supply: Incorporating a labour-leisure choice transforms the model, adding a control variable and making the income process endogenous. This is a critical step towards solving full-scale DSGE models.
3. Confronting the Curse of Dimensionality: The final and most significant stage is to deploy the Neural Network solvers against model classes where traditional global methods like Value Function Iteration (VFI) and Endogenous Grid Method (EGM) become computationally prohibitive or intractable. The key question is whether the NN's mesh-free, function-approximation approach can maintain accuracy where grid-based methods fail. Prime candidates include:

https://doi.org/10.26219/heal.aueb.9687

a. Heterogeneous Agent New Keynesian (HANK) models: These models, which combine heterogeneous households with nominal rigidities, are the frontier of modern macroeconomics but are notoriously difficult to solve due to the need to track the entire wealth distribution—a high-dimensional object. Neural Networks, trained to learn the aggregate law of motion or the policy functions of a subset of agents, could offer a transformative leap in efficiency.
b. Multi-country or multi-sector models: Models with several interconnected sectors or countries involve a state space that grows exponentially with the number of entities. This is a classic manifestation of the curse of dimensionality that Neural Networks are theoretically well-suited to mitigate.
c. Models with portfolio choice: Introducing multiple asset classes adds a discrete or continuous choice dimension that drastically increases the complexity of the problem, another area where traditional global solvers struggle.

By following this pathway, future research can systematically evaluate whether the promise shown by Neural Networks in this simple setting scales to the most challenging problems in the field. The ultimate goal is not merely to replicate existing solutions but to enable the solution of models that are currently beyond the reach of conventional methods, thereby expanding the frontier of what can be quantitatively analysed in macroeconomics.

9.2 Advancing the Neural Network Methodology

The current architecture is effective but primary. Significant gains in efficiency and accuracy may be unlocked by incorporating modern deep learning advances.

- Parameterized Solutions: As previously discussed, a pivotal innovation would be to train Neural Networks to be functions of both the state variables and the deep parameters of the model (e.g., β, γ, ρ). A single network trained over a plausible range of parameter values would function as a "meta-solver," eliminating the need for retraining during estimation and allowing for instant counterfactual policy analysis across different economic structures.
- Alternative Architectures and Loss Functions: The standard feedforward network is just one possibility. Architectures like Physics-Informed Neural Networks (PINNs) could more naturally encode the model's equilibrium constraints. Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks might be better suited for learning the temporal dynamics inherent in time-series simulation. Furthermore, the

 https://doi.org/10.26219/heal.aueb.9687

loss function could be refined to place greater weight on economically relevant regions or to better enforce model constraints.

- Active Learning and Adaptive Sampling: Instead of using fixed datasets of uniform and correlated samples, the training process could be made more efficient through active learning. The network could iteratively query the states where its current Euler or Bellman residual is highest, focusing its learning effort on the most challenging parts of the state space.

9.3 New Applications and Integration

Beyond simply solving models faster, Neural Network solvers enable entirely new workflows and analyses.

- Estimation at Scale: The "solve-once, simulate-instantly" property of a parameterized Neural Network solver could revolutionize Bayesian estimation of medium- to high-dimensional DSGE models, turning a computationally prohibitive process into a tractable one.
- Real-Time Policy Analysis: A pre-trained, parameterized network could be deployed in a policy dashboard, allowing central bankers or fiscal authorities to run instant counterfactual simulations for different policy rules or shock scenarios.
- Bridging to Machine Learning for Inference: A seamless pipeline could be developed where a Neural Network solves the model and then another machine learning model is used for indirect inference or to match simulated moments to data, fully leveraging the different strengths of these computational tools.

9.4 Conclusion

This thesis has taken a necessary first step, providing a rigorous proof-of-concept by successfully applying Neural Networks to the simplest representative agent model. The results are certainly promising, evidence that these methods can achieve global accuracy and offer a unique computational advantage for large-scale simulation. The path forward is now clear and compelling. The methodology developed here does not represent an end point but rather a foundational blueprint. By systematically escalating model complexity, refining the Neural Network architectures, and exploring novel applications, future research can build upon this foundation to tackle the most computationally intensive problems in modern macroeconomics. The integration of machine learning into the computational economist's toolkit is not a distant

https://doi.org/10.26219/heal.aueb.9687

future prospect; it is an ongoing evolution, and this work provides a solid and encouraging starting point for that journey.

Chapter 10. Conclusion

This thesis has provided a comprehensive evaluation of local, global, and machine-learning–based numerical methods for solving stochastic Real Business Cycle (RBC) models. Through a unified calibration and consistent evaluation framework, it has demonstrated how each method performs across increasing levels of volatility and shock intensity, thereby quantifying the balance between computational efficiency and global accuracy.

The findings confirm a clear hierarchy of applicability.

Perturbation methods, both first- and second-order, are highly efficient for small shocks and near-steady-state dynamics, offering immediate analytical insights. However, their local nature limits their reliability once the economy is exposed to stronger nonlinearities or larger volatility. The stress-test analysis established explicit quantitative thresholds beyond which Perturbation accuracy deteriorates, providing a concrete guide for practitioners on when to switch to global methods.

Among global solvers, the Endogenous Grid Method (EGM) emerged as the most balanced and robust technique. It replicates the accuracy of the benchmark Value Function Iteration (VFI) while requiring only a fraction of the computational cost, making it ideal for most practical applications and policy exercises. Its performance across all volatility regimes confirms its status as the preferred benchmark for global analysis in small- to medium-scale models.

The integration of Neural Networks represents the most novel contribution of this work. Both the distillation-based NN(Global) and the purely equation-driven NN(Equation) attained near-benchmark precision, validating their ability to learn and reproduce equilibrium dynamics. Despite their high training cost, their ability to evaluate policies in microseconds offers transformative potential for large-scale simulations, estimation, and real-time policy evaluation. The equation-based variant, in particular, demonstrated superior generalization in extreme regions of the state space, confirming that deep learning can recover globally consistent economic solutions directly from structural conditions.

Overall, the comparative analysis underscores that there is no universally superior method; rather, the optimal solver depends on the research objective. Perturbation remains optimal for local analysis and calibration, the Endogenous Grid Method (EGM) provides the most practical

https://doi.org/10.26219/heal.aueb.9687

global benchmark, and Neural Networks offer an emerging frontier for large-scale or high-dimensional environments where traditional solvers become infeasible. The methodological framework and empirical evidence presented here thus position Neural Networks not as a theoretical curiosity but as a ready-to-use, globally accurate computational tool for modern macroeconomics.

Chapter 11. Bibliography

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

Fernández-Villaverde, J., Gosling, D., Kuester, K., & Rubio-Ramírez, J. F. (2020). *Solving high-dimensional dynamic programming problems using deep learning*. NBER Working Paper No. 28298.

Heer, B., & Maussner, A. (2009). *Dynamic General Equilibrium Modeling: Computational Methods and Applications* (2nd ed.). Springer.

Judd, K. L. (1998). *Numerical Methods in Economics*. MIT Press.

Judd, K. L., Maliar, L., Maliar, S., & Valero, R. (2017). *How to solve dynamic stochastic models computing expectations just once*. Quantitative Economics, 8(3), 851–893.

Kase, M. (2022). *Solving DSGE models with neural networks*. Journal of Economic Dynamics and Control, 144, 104528.

Maliar, L., & Maliar, S. (2014). *Merging simulation and projection approaches to solve high-dimensional problems with an application to a new Keynesian model*. Quantitative Economics, 5(1), 1–46.

Maliar, L., Maliar, S., & Winant, P. (2021). *Deep learning for solving dynamic economic models*. Journal of Monetary Economics, 124, 1–19.

Miranda, M. J., & Fackler, P. L. (2002). *Applied Computational Economics and Finance*. MIT Press.

Schmitt-Grohé, S., & Uribe, M. (2004). *Solving dynamic general equilibrium models using a second-order approximation to the policy function*. Journal of Economic Dynamics and Control, 28(4), 755–775.

Sims, C. A. (2002). *Solving linear rational expectations models*. Computational Economics, 20(1–2), 1–20.

Taylor, J. B., & Uhlig, H. (1990). *Solving nonlinear stochastic growth models: A comparison of alternative solution methods*. Journal of Business & Economic Statistics, 8(1), 1–17.

https://doi.org/10.26219/heal.aueb.9687

Uhlig, H. (1999). *A toolkit for analyzing nonlinear dynamic stochastic models easily*. In R. Marimon & A. Scott (Eds.), *Computational Methods for the Study of Dynamic Economies* (pp. 30–61). Oxford University Press.