Board of Governors of the Federal Reserve System
International Finance Discussion Papers
Number 920
February 2008

# On the Application of Automatic Differentiation to the Likelihood Function for Dynamic General Equilibrium Models

Houtan Bastani
Luca Guerrieri

# On the Application of Automatic Differentiation to the Likelihood Function for Dynamic General Equilibrium Models*

Houtan Bastani and Luca Guerrieri**
Federal Reserve Board

February, 2008

**Abstract**

A key application of automatic differentiation (AD) is to facilitate numerical optimization problems. Such problems are at the core of many estimation techniques, including maximum likelihood. As one of the first applications of AD in the field of economics, we used Tapenade to construct derivatives for the likelihood function of any linear or linearized general equilibrium model solved under the assumption of rational expectations. We view our main contribution as providing an important check on finite-difference (FD) numerical derivatives. We also construct Monte Carlo experiments to compare maximum-likelihood estimates obtained with and without the aid of automatic derivatives. We find that the convergence rate of our optimization algorithm can increase substantially when we use AD derivatives.

**Keywords**: General Equilibrium Models, Kalman Filter, Maximum Likelihood
**JEL Classification**: C52, C61, C63

* The authors thank Chris Gust, Alejandro Justiniano, and seminar participants at the 2007 meeting of the Society for Computational Economics. We are particularly grateful to Gary Anderson for countless helpful conversations, as well as for generously sharing his programs with us. The views expressed in this paper are solely the responsibility of the authors and should not be interpreted as reflecting the views of the Board of Governors of the Federal Reserve System or of any other person associated with the Federal Reserve System.

** Corresponding author. Telephone (202) 452-2550. E-mail Luca.Guerrieri@frb.gov.

# 1 Introduction

While applications of automatic differentiation have spread across many different disciplines, they have remained less common in the field of economics.[1] Based on the successes reported in facilitating optimization exercises in other disciplines, we deployed AD techniques to assist with the estimation of dynamic general equilibrium (DGE) models. These models are becoming a standard tool that central banks use to inform monetary policy decisions. However, the estimation of these models is complicated by the many parameters of interest. Thus, typically, the optimization method of choice makes use of derivatives. However, the complexity of the models does not afford a closed-form representation for the likelihood function. Finite-difference methods have been the standard practice to obtain numerical derivatives in this context. Using Tapenade (see (Hascoët 2004), (Hascoët, Greborio, and Pascual 2005), (Hascoët, Pascual, and Dervieux 2005)), we constructed derivatives for a general formulation of the likelihood function, which takes as essential input the linear representation of the model's conditions for an equilibrium.

The programming task was complicated by the fact that the numerical solution of a DGE model under rational expectations relies on fairly complex algorithms.[2] We use Lapack routines for the implementation of the solution algorithm. In turn, our top Lapack routines make use of a large number of Blas routines. A byproduct of our project has been the implementation of numerous AD derivatives of the double precision subset of Blas routines. Table 1 lists the routines involved.

In the remainder of this paper, Section 2 lays out the general structure of a DGE model and describes our approach to setting up the model's likelihood function.

[1] Examples of AD contributions to the computational finance literature are (C. H. Bischof 2002), (M. Giles 2006), (Giles 2007).

[2] In this paper we focus on the first-order approximation to the solution of a DGE model. Many alternative approaches have been advanced. We use the algorithm described by (Anderson and Moore 1985) which has the marked advantage of not relying on complex decompositions.

Section 3 outlines the step we took to implement the AD derivatives and how we built confidence in our results. Section 4 gives an example of a DGE model that we used to construct Monte Carlo experiments to compare maximum-likelihood estimates that rely, alternatively, on AD or FD derivatives, reported in Section 5. Section 6 concludes.

# 2 General Model Description and Estimation Strategy

The class of DGE models that is the focus of this paper take the general form:

$$H(\theta)\begin{pmatrix} E_t X_{t+1} \\ X_t \\ X_{t-1} \end{pmatrix} = 0. \tag{1}$$

In the equation above, $H$ is a matrix whose entries are a function of the structural parameter vector $\theta$, while $X_t$ is a vector of the model's variables (including the stochastic innovations to the shock processes). The term $E_t$ is an expectation operator, conditional on information available at time $t$ and the model's structure as in equation 1. Notice that allowing for only one lead and one lag of $X_t$ in the above equation implies no loss of generality.

The model's solution takes the form:

$$X_t = S(H(\theta))X_{t-1}, \tag{2}$$

thus, given knowledge of the model's variables at time $t-1$, a solution determines the model's variables at time $t$ uniquely. The entries of the matrix $S$ are themselves functions of the matrix $H$ and, in turn, of the parameter vector $\theta$.

Partitioning $X_t$ such that $X_t = \begin{pmatrix} x_t \\ \epsilon_t \end{pmatrix}$, where $\epsilon_t$ is a collection of all the innovations to the exogenous shock processes (and possibly rearranging the system) it

is convenient to rewrite the model's solution as

$$x_t = A(H(\theta))x_{t-1} + B(H(\theta))\epsilon_t. \tag{3}$$

Again, the entries in the matrices $A$ and $B$ are fundamentally functions of the parameter vector $\theta$. Given a subset of the entries in $x_t$ as observable, call these entries $y_t$, the state-space representation of the system takes the form:

$$x_t = A(H(\theta))x_{t-1} + B(H(\theta))\epsilon_t \tag{4}$$

$$y_t = Cx_t \tag{5}$$

Without loss of generality, we restrict the matrix $C$ to be a selector matrix, which picks the relevant entries of $x_t$. Using the Kalman Filter recursions, we can express the likelihood function for the model as:

$$L = L(A(\theta), B(\theta), C, y_{t-h}, ..., y_t) \tag{6}$$

where $y_{t-h}$ and $y_t$ are respectively the first and last observation points available.

The routines we developed, given an input $H(\theta)$, produce the derivative of the likelihood function with respect to the structural parameters, $\frac{\partial L}{\partial \theta}$, and as an intermediate product, $\frac{\partial A}{\partial \theta}$, the derivative of the model's reduced-form parameters with respect to the structural parameters.

# 3 Implementing AD Derivatives

To obtain AD derivatives of the likelihood function, we used Tapenade in tangent mode. Tapenade required limited manual intervention on our part. This is remarkable given that the code to be differentiated consisted of approximately 80 subroutines for a total of over 17,000 lines of code. The derivative-augmented code produced by Tapenade covers approximately 25,000 lines (the original code has a size of 554 kilobytes and the differentiated code is 784 kilobytes in size).

Recoding became necessary when the Lapack or Blas routines we called did not explicitly declare the sizes of the arguments in the calling structure and instead

allowed for arbitrary sizing (possibly exceeding the storage requirements). A more limited recoding was required when we encountered the use of "GOTO" statements in the Fortran 77 code of the Blas library, which Tapenade could not process.

More substantially, two of the decompositions involved in the model solution, the real Schur decomposition and the singular-value decomposition, are not always unique. Parametric restrictions of the models we tested could ensure uniqueness of these decompositions. In those cases, we verified that AD derivatives obtained through Tapenade satisfied some basic properties of the decompositions that we derived analytically, but our test failed whenever we relaxed those parametric restrictions to allow for more general model specifications

In particular, we relied on the Lapack routine DGEESX to implement the real Schur decomposition. For a given real matrix $E$, this decomposition produces a unitary matrix $X$, such that $T = X^H E X$ is quasitriangular. Given $\frac{\partial E}{\partial \theta}$, we need that the derivative $\frac{\partial X}{\partial \theta}$ satisfy $\frac{\partial T}{\partial \theta} = \frac{\partial X^H}{\partial \theta} E X + X^H \frac{\partial E}{\partial \theta} X + X^H E \frac{\partial X}{\partial \theta}$, where $\frac{\partial T}{\partial \theta}$ is itself quasitriangular. This property failed to be met by our AD derivatives when our choice of E implied a non-unique Schur decomposition. To obviate this problem, we substituted the AD derivative for the DGEESX routine with the analytical derivative of the Schur decomposition as outlined in (Anderson 1987).

Similarly, the singular value decomposition, implemented through the DGESVD routine in the Lapack library, given a real matrix $E$, produces unitary matrices $U$ and $V$ and a diagonal matrix $D$, such that $E = UDV^T$. Given $\frac{\partial E}{\partial \theta}$, it can be shown that $U^T \frac{\partial E}{\partial \theta} V = U^T \frac{\partial U}{\partial \theta} D + \frac{\partial D}{\partial \theta} + D \frac{\partial V}{\partial \theta} V$, where $\frac{\partial D}{\theta}$ is diagonal and $U^T \frac{\partial U}{\partial \theta}$ and $\frac{\partial V}{\partial \theta} V$ are both antisymmetric. Our AD derivative of the routine DGESVD failed to satisfy this property when the matrix $E$ had repeated singular values (making the decomposition non-unique). We substituted our AD derivative with the analytical derivative derived by (Papadopoulo and Lourakis 2000).

To test the derivative of the likelihood function, we used a two-pronged approach. For special cases of our model that could be simplified enough as to yield a closed-

form analytical solution, we computed analytical derivatives and found them in agreement with our AD derivatives, accounting for numerical imprecision. To test the derivatives for more complex models that we could not solve analytically, we relied on comparisons with centered FD derivatives. Generally with a step size of $10^{-8}$ we found broad agreement between our AD derivatives and FD derivatives. Plotting AD and FD side by side, and varying the value at which the derivatives were evaluated, we noticed that the FD derivatives appeared noisier than the AD derivatives. We quantify the "noise" we observed in an example below.

# 4 Example Application

As a first application of our derivatives, we consider a real business cycle model augmented with sticky prices and sticky wages, as well as several real rigidities, following the work of (Smets and Wouters 2003). Below, we give a brief description of the optimization problems solved by agents in the model, which allows us to interpret the parameters estimated in the Monte Carlo exercises that follow.

There is a continuum of households of measure 1, indexed by $h$, whose objective is to maximize a discounted stream of utility according to the following setup:

$$\begin{aligned}
&\max_{[C_t(h),W_t(h),I_t(h),K_{t+1}(h),B_{t+1}(h)]} E_t \sum_{j=0}^{\infty} \beta^j \left(U(C_{t+j}(h), C_{t+j-1}(h))\right. \\
&+V(L_{t+j}(h))) + \beta^j \lambda_{t+j}(h) \left[\Pi_t(h) + T_{t+j}(h) + (1-\tau_{Lt}) W_{t+j}(h) L_{t+j}(h)\right. \\
&+(1-\tau_{Kt}) R_{kt+j} K_{t+j}(h) - \frac{1}{2}\psi_I P_{t+j} \frac{\left(I_{t+j}(h) - I_{t+j-1}(h)\right)^2}{I_{t+j-1}(h)} \\
&\left.-P_{t+j} C_{t+j}(h) - P_{t+j} I_{t+j}(h) - \int_s \psi_{t+j+1,t+j} B_{t+j+1}(h) + B_{t+j}(h)\right] \\
&+\beta^j Q_{t+j}(h) \left[(1-\delta) K_{t+j}(h) + I_{t+j}(h) - K_{t+j+1}(h)\right].
\end{aligned}$$

The utility function depends on consumption $C_t(h)$ and labor supplied $L_t(h)$. The parameter $\beta$ is a discount factor for future utility. Households choose streams for consumption $C_t(h)$, wages $W_t(h)$, investment $I_t(h)$, capital $K_{t+1}(h)$ and bond holdings $B_{t+1}(h)$, subject to the budget constraint, whose Lagrangian multiplier is $\lambda_t(h)$,

capital accumulation equation, whose Lagrangian multiplier is $Q_t(h)$, and the labor demand schedule $L_t(h) = L_t \left(\frac{W_t(h)}{W_t}\right)^{-\frac{1+\theta_w}{\theta_w}}$. Households rent to firms (described below) both capital, at the rental rate $R_{Kt}$, and labor at the rental rate $W_t(h)$, subject to labor taxes at the rate $\tau_{Lt}$ and to capital taxes at the rate $\tau_{Kt}$. There are quadratic adjustment costs for investment, governed by the parameter $\psi_I$, and capital depreciates at a per-period rate $\delta$. We introduce Calvo-type contracts for wages following (Erceg, Henderson, and Levin 2000). According to these contracts, the ability to reset wages for a household $h$ in any period $t$ follows a Poisson distribution. A household is allowed to reset wages with probability $1 - \xi_w$. If the wage is not reset, it is updated according to $W_{t+j}(h) = W_t(h)\pi^j$ (where $\pi$ is the steady-state inflation rate), as in (Yun 1996). Finally, $T_t(h)$ and $\Pi_t(h)$ represent, respectively, net lump-sum transfers from the government and an aliquot share of the profits of firms.

In the production sector, we have a standard Dixit-Stiglitz setup with nominal rigidities. Competitive final producers aggregate intermediate products for resale. Their production function is

$$Y_t = \left[\int_0^1 Y_t(f)^{\frac{1}{1+\theta_p}}\right]^{1+\theta_p} \tag{7}$$

and from the zero profit condition the price for final goods is

$$P_t = \left[\int_0^1 P_t(f)^{-\frac{1}{\theta_p}}\right]^{-\theta_p}. \tag{8}$$

where $P_t(f)$ is the price for a unit of output for the intermediate firm $f$.

Intermediate firms are monopolistically competitive. There is complete mobility of capital and labor across firms. Their production technology is given by

$$Y_t(f) = A_t K_t(f)^{\alpha} L_t^d(f)^{1-\alpha}. \tag{9}$$

Intermediate firms take input prices as given. $L_t^d(f)$, which enters the intermediate firms' production function, is an aggregate over the skills supplied by each household,

and takes the form $L_t^d(f) = \left(\int_h L_t(h)^{\frac{1}{1+\theta_w}}\right)^{1+\theta_w}$. $A_t$ is the technology level and evolves according to an autoregressive (AR) process:

$$A_t - A = \rho_A \left(A_{t-1} - A\right) + \epsilon_{At}, \tag{10}$$

where $\epsilon_{At}$ is an *iid* innovation with standard deviation $\sigma_A$, and $A$ is the steady-state level for technology. Intermediate firms set their prices $P_t(f)$ according to Calvo-type contracts with reset probabilities $1 - \psi_P$. When prices are not reset, they are updated according to $P_{t+j}t(f) = P_t(f)\pi^j$.

Finally, the government sector sets a nominal risk-free interest rate according to the reaction function:

$$i_t = \frac{\pi}{\beta} - 1 + \gamma_\pi(\pi_t - \pi) + \gamma_Y(log(Y_t) - log(Y_{t-1}) + \epsilon_{it}, \tag{11}$$

where inflation $\pi_t \equiv \frac{P_t}{P_{t-1}}$, and $\epsilon_{it}$ is itself an AR process of order 1. For this process, we denote the AR coefficient with $\rho_i$; the stochastic innovation is iid with standard deviation $\sigma_i$. Notice that, in this setting, households are Ricardian, hence the time-profile of net lump-sum transfers is not distortionary. We assume that these transfers are set according to:

$$\tau_{Lt} W_t L_t + \tau_{Kt} R_{Kt} K_t = G_t + T_t. \tag{12}$$

Labor taxes, $\tau_{Lt}$, and capital taxes, $\tau_{Kt}$, follow exogenous AR processes

$$\tau_{Lt} - \tau_L = \rho_L \left(\tau_{Lt-1} - \tau_L\right) + \epsilon_{Lt}, \tag{13}$$

$$\tau_{Kt} - \tau_K = \rho_K \left(\tau_{Kt-1} - \tau_K\right) + \epsilon_{Kt}, \tag{14}$$

as does Government spending (expressed as a share of output)

$$\frac{G_t}{Y_t} - \frac{G}{Y} = \rho_G \left(\frac{G_{t-1}}{Y_{t-1}} - \frac{G}{Y}\right) + \epsilon_{Gt}. \tag{15}$$

In the equations above, the exogenous innovations $\epsilon_{Lt}, \epsilon_{Kt}, \epsilon_{Gt}$ are *iid* with standard deviations $\sigma_L$, $\sigma_{Kt}$, and $\sigma_G$, respectively. The parameters $\tau_L$, $\tau_K$, and $\frac{G}{Y}$, without a time subscript, denote steady-state levels.

The calibration strategy follows (Erceg, Guerrieri, and Gust 2005) and parameter values are reported in Table 2. By linearizing the necessary conditions for the solution of the model, we can express them in the format of Equation (1).

# 5 Monte Carlo Results

Using the model described in Section 4 as the data-generating process, we set up a Monte Carlo experiment to compare maximum-likelihood estimates obtained through two different optimization methods. One of the methods relies on our AD derivative of the model's likelihood function. The alternative method, uses a two-point, centered, finite-difference approximation to the derivative.

In setting up the likelihood function, we limit our choices for the observed variables in the vector $y_t$ of equation (5) to four series, namely: growth rate of output $\log(Y_t) - log(Y_{t-1})$, price inflation $\pi_t$, wage inflation $\omega_t \equiv \frac{W_t}{W_{t-1}}$, and the policy interest rate $i_t$. For each Monte Carlo sample, we generate 200 observations, equivalent to 50 years of data given our quarterly calibration, a sample length often used in empirical studies. We attempt to estimate the parameters $\rho_i$, $\sigma_i$, governing the exogenous shock process for the interest rate reaction function; $\psi_P$, $\psi_W$, the Calvo contract parameters for wages and prices; and $\gamma_\pi$, and $\gamma_Y$ the weights in the monetary policy reaction function for inflation and activity. In the estimation exercises, we kept the remaining parameters at their values in the data-generating process as detailed in Table 2. We considered 1,000 Monte Carlo samples.[3] The two experiments described below differ only insofar as we chose two different initialization points for the optimization routines we used to maximize the likelihood function.

Figure 1 shows the sampling distribution for the parameter estimates from our

[3]Our maximum-likelihood estimates were constructed using the MATLAB optimization routine FMINUNC. When the optional argument "LargeScale" is set to "OFF", this routine uses a limited memory quasi-Newton conjugate gradient method, which takes as input first derivatives of the objective function, or an acceptable FD approximation.

Monte Carlo exercise when we initialize the optimization routine at the true parameter values used in the data-generating process. The black bars in the various panels denote the estimates that rely on AD derivatives, while the white bars denote the estimates obtained with FD derivatives. The optimization algorithm converged for all of the 1,000 Monte Carlo samples.[4] We verified that the optimization routine did move away from the initial point towards higher likelihood values, so that clustering of the estimates around the truth do not merely reflect the initialization point. For our experiment, the figure makes clear that when the optimization algorithm is initiated at the true value for the parameters of interest, reliance on FD derivatives minimally affects the maximum-likelihood estimates for those parameters.[5]

Of course, the true value of the parameters do not necessarily coincide with the ML parameter estimates for small samples. Yet, it is unrealistic to assume that a researcher would happen on such good starting values. Figure 2 reports the sampling distribution of estimates obtained when we initialize the optimization algorithm at arbitrary values for the parameters being estimated, away from their true values. For the estimates reported in Figure 2, we chose $\rho_i = 0.6$, $\sigma_i = 0.4$, $\psi_P = .5$, $\psi_W = 0.5$, $\gamma_\pi = 3$, $\gamma_Y = 0.15$. The bars in Figure 2 show the frequency of estimates in a given range as a percentage of the 1,000 experiments we performed. We excluded results for which our optimization algorithm failed to converge. The figure makes clear that the convergence rate is much higher when using AD derivatives (47.2% instead of 28.3% for FD derivatives). Moreover, it is also remarkable that the higher convergence rate is not accompanied by a deterioration of the estimates (the

[4]For our MATLAB optimization routine, we set the convergence criterion to require a change in the objective function smaller than to $10^{-4}$, implying 6 significant figures for our specific likelihood function. This choice seemed appropriate given the limited precision of observed series in practical applications.

[5]We experimented with a broad set of Monte Carlo experiments by varying the choice of estimation parameters, so as to encompass the near totality of parameters in the calibration table, or so as to study individual parameters in isolation. We found results broadly in line with the particular Monte Carlo experiments we are reporting below. Our results also appear robust to broad variation in the calibration choice.

increased height of the black bars in the figure is proportional to that of the white bars).

To quantify the difference between AD and FD derivatives of the likelihood function for one of our samples, we varied the parameters we estimated one at a time. Figure 3 shows the percentage difference in the magnitude of the AD and FD derivatives for $\rho_i$ and $\sigma_i$. We discretized the ranges shown using a grid of 1,000 equally spaced points. The differences are generally small percentage-wise, although, on occasion, they spike up, or creep up as we move away from the true value, as in the case of $\sigma_i$. For the other parameters we estimated, we did not observe differences in the magnitudes of the AD and FD derivatives larger than $10^{-4}$ over ranges consistent with the existence of a rational expectations equilibrium for our model.

# 6 Conclusion

Given that the approximation error for a first derivative of the likelihood function of a DGE model computed through FD methods depends on the size of the second derivative, which itself is subject to approximation error, we view having an independent check in the form of automatic derivatives as a major contribution of our work. As an example application, we showed that AD derivatives can facilitate the computation of maximum-likelihood estimates for the parameters of a DGE model.

# References


Anderson, G. (1987). A Procedure for Differentiating Perfect-Foresight-Model Reduced-Form Coefficients. *Journal of Economic Dynamics and Control 11*, 465–81.

Anderson, G. and G. Moore (1985). A Linear Algebraic Procedure for Solving Linear Perfect Foresight Models. *Economic Letters 17*, 247–52.

C. H. Bischof, H. M. Bücher, B. L. (2002). Automatic Differentiation for Computational Finance. In B. R. E. J. Kontoghiorghes and S. Siokos (Eds.), *Computational Methods in Decision-Making*. springer.

Erceg, C. J., L. Guerrieri, and C. Gust (2005). Can Long-Run Restrictions Identify Technology Shocks? *Journal of the European Economic Association 3*(6), 1237–1278.

Erceg, C. J., D. W. Henderson, and A. T. Levin (2000). Optimal monetary policy with staggered wage and price contracts. *Journal of Monetary Economics 46*(2), 281–313.

Giles, M. (2007). Monte Carlo Evaluation of Sensitivities in Computational Finance. In E. A. Lipitakis (Ed.), *HERCMA The 8th Hellenic European Research on Computer Mathematics and its Applications Conference*.

Hascoët, L. (2004). TAPENADE: a tool for Automatic Differentiation of programs. In *Proceedings of $4^{th}$ European Congress on Computational Methods, ECCOMAS'2004, Jyvaskyla, Finland*.

Hascoët, L., R.-M. Greborio, and V. Pascual (2005). *Computing Adjoints by Automatic Differentiation with TAPENADE*. Springer. Forthcoming.

Hascoët, L., V. Pascual, and D. Dervieux (2005). *Automatic Differentiation with TAPENADE*. Springer. Forthcoming.

M. Giles, P. G. (2006). Smoking Adjoints: Fast Monte Carlo Greeks. *Risk Magazine 19*, 88–92.

Papadopoulo, T. and M. I. A. Lourakis (2000). Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications. *Lecture Notes in Computer Science 1842/2000*, 554–570.

Smets, F. and R. Wouters (2003). An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area. *Journal of the European Economic Association 1*(5), 1123–1175.

Yun, T. (1996). Nominal price rigidity, money supply endogeneity, and business cycles. *Journal of Monetary Economics 37*, 345–370.

Table 1: List of library functions

| *Blas Functions* | | | | | | | |
|---|---|---|---|---|---|---|---|
| daxpy.f | dcopy.f | ddot.f | dgemm.f | dgemv.f | dger.f | dnrm2.f | drot.f |
| dscal.f | dswap.f | dtrmm.f | dtrmv.f | dtrsm.f | | | |
| *Lapack Functions* | | | | | | | |
| dgebak.f | dgebal.f | dgeesx.f | dgehd2.f | dgehrd.f | dgeqp3.f | dgeqr2.f | dgeqrf.f |
| dgesv.f | dgetf2.f | dgetrf.f | dgetrs.f | dhseqr.f | dlacn2.f | dlacpy.f | dladiv.f |
| dlaexc.f | dlahqr.f | dlahr2.f | dlaln2.f | dlange.f | dlanv2.f | dlapy2.f | dlaqp2.f |
| dlaqps.f | dlaqr0.f | dlaqr1.f | dlaqr2.f | dlaqr3.f | dlaqr4.f | dlaqr5.f | dlarfb.f |
| dlarf.f | dlarfg.f | dlarft.f | dlarfx.f | dlartg.f | dlascl.f | dlaset.f | dlassq.f |
| dlaswp.f | dlasy2.f | dorg2r.f | dorghr.f | dorgqr.f | dorm2r.f | dormqr.f | dtrexc.f |
| dtrsen.f | dtrsyl.f | dtrtrs.f | | | | | |

Table 2: Calibration

| **Parameter** | **Used to Determine** | **Parameter** | **Used to Determine** |
|---|---|---|---|
| *Parameters governing households' and firms' behavior* | | | |
| $\beta = 0.997$ | discount factor | $\phi_I = 3$ | investment adj. cost |
| $\tau_L = 0.28$ | steady state labor tax rate | $\tau_K = 0$ | steady state capital tax rate |
| $\psi_P = 0.75$ | Calvo price parameter | $\psi_W = 0.75$ | Calvo wage parameter |
| $\delta = 0.025$ | depreciation rate | | |
| *Monetary Policy Reaction Function* | | | |
| $\gamma_\pi = 1.5$ | inflation weight | $\gamma_Y = 0.5$ | output weight |
| *Exogenous Processes* | | | |
| | *AR(1) Coefficient* | | *Standard Deviation* |
| $\rho_L = 0.98$ | labor tax rate | $\sigma_L = 3.88$ | labor tax rate innovation |
| $\rho_K = 0.97$ | capital tax rate | $\sigma_K = 0.80$ | capital tax innovation |
| $\rho_G = 0.98$ | govt spending | $\sigma_G = 0.30$ | govt spending innovation |
| $\rho_i = 0.95$ | monetary policy | $\sigma_i = 0.11$ | monetary policy innovation |
| $\rho_A = 0.95$ | technology | $\sigma_A = 0.94$ | labor tax innovation |

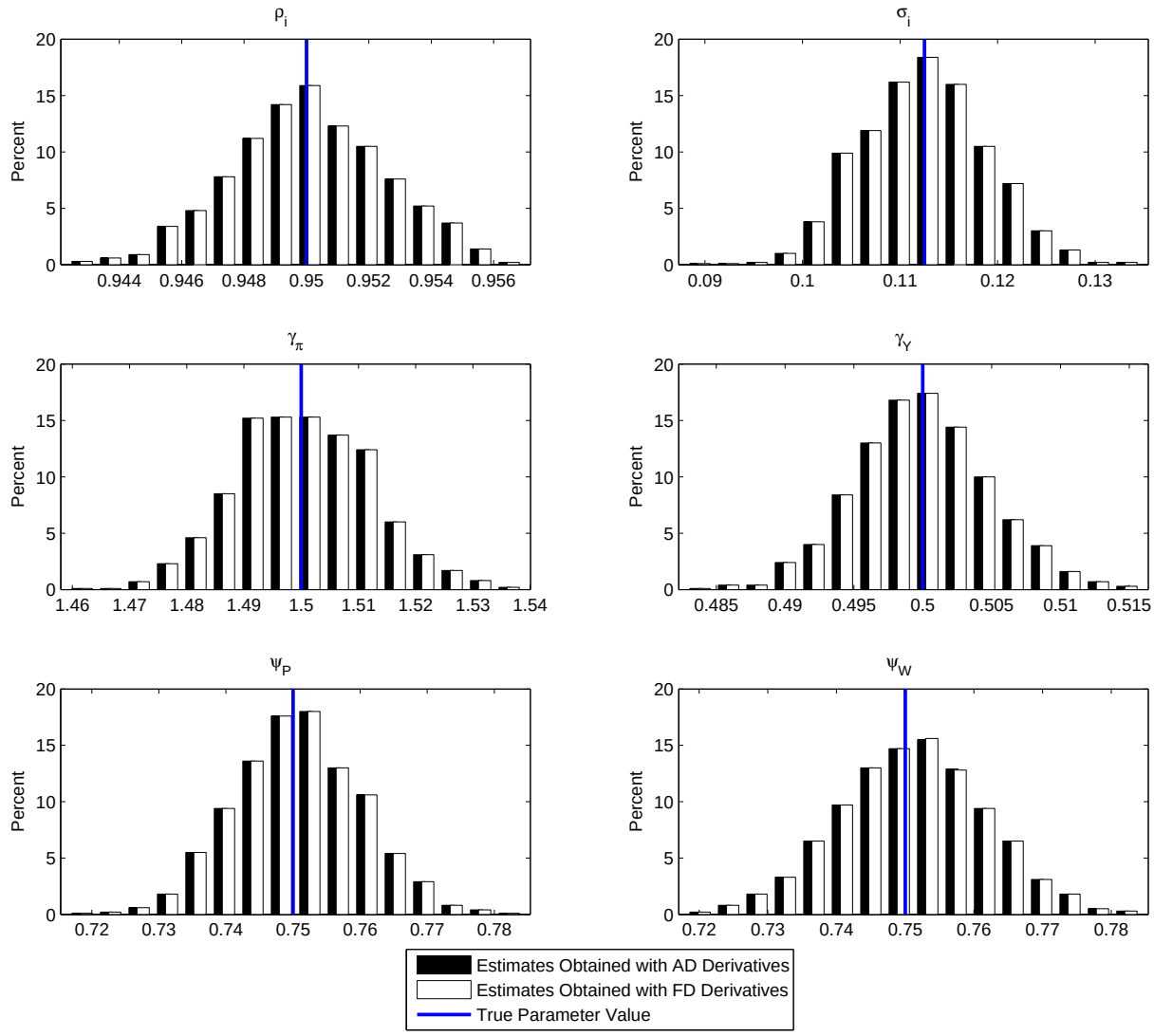


Figure 1: Sampling Distribution of Parameter Estimates; the Initial Guesses Coincided with the True Values in the Data-Generating Process.

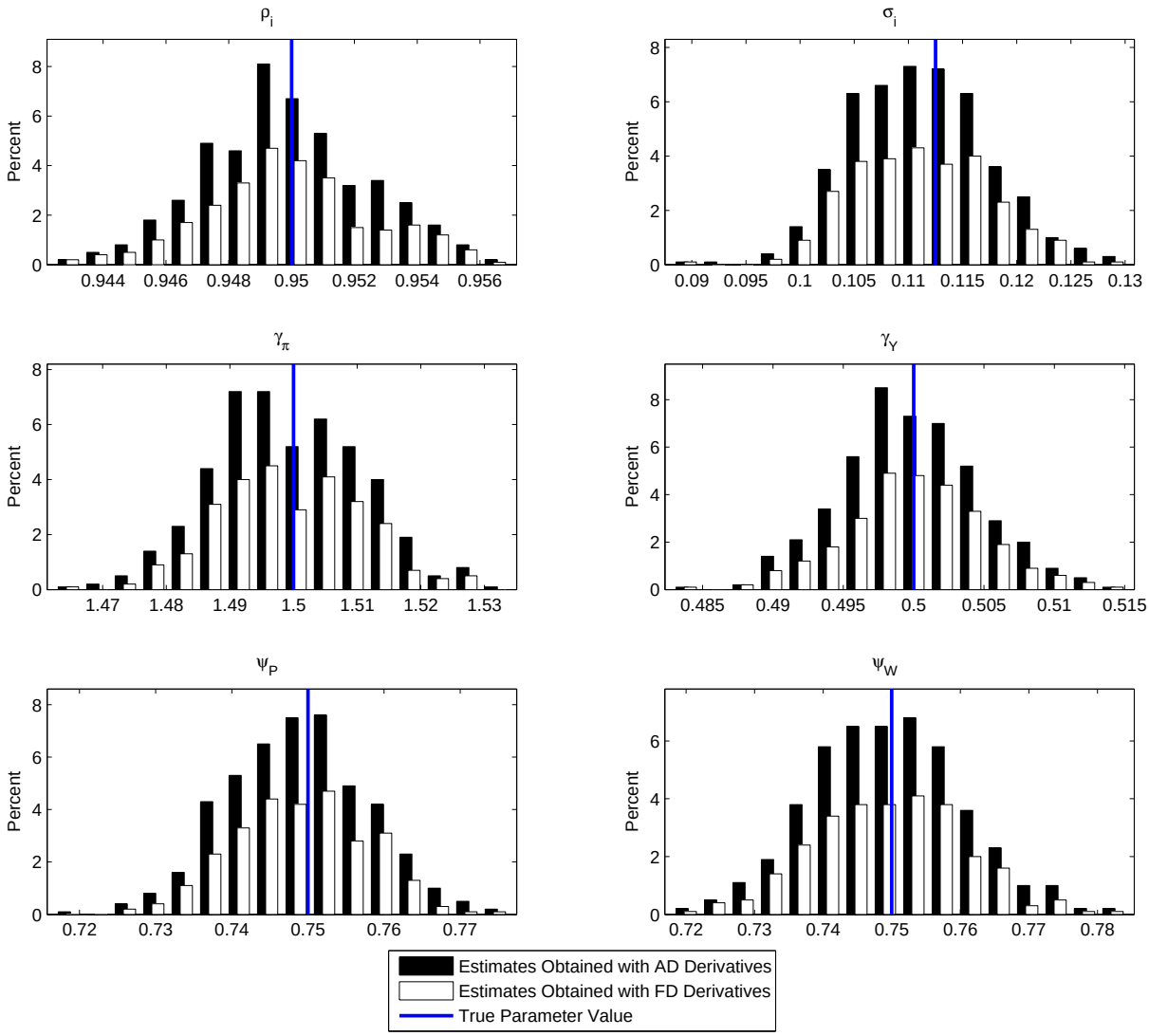


Figure 2: Sampling Distribution of Parameter Estimates; the Initial Guesses Did Not Coincide with the True Values in the Data-Generating Process.

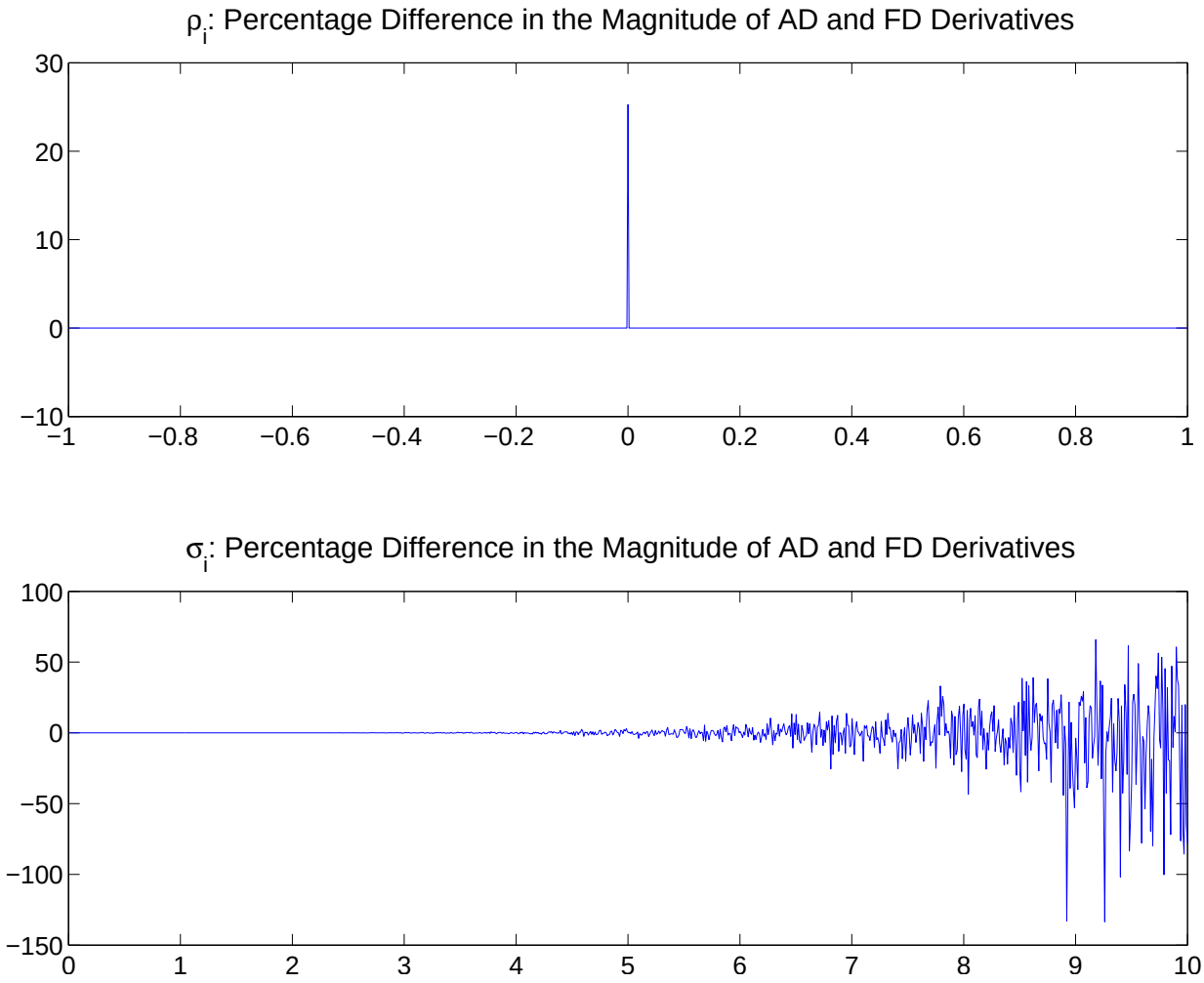


Figure 3: Percentage Difference Between AD and FD Derivatives.