SSASIM Guide

Martin R. Holmer
Policy Simulation Group

TEL 202-526-0406
NET martin.holmer@gmail.com
WEB www.polsim.com

February 2016

Preface

This is an intermediate guide for policy analysts using SSASIM, a social security and accounts policy simulation model. This guide assumes that the analyst has already read both the *Getting Started with the PSG Models* ⟨http://www.polsim.com/psghelp.pdf⟩ and the *Getting Started with the RSF Toolkit* ⟨http://www.polsim.com/rsfhelp.pdf⟩ documents.

SSASIM was originally developed under a series of contracts from SSA as part of the 1994–96 Advisory Council's activities, and has been enhanced subsequently with support from AARP's Public Policy Institute, SSA's Office of Policy, EBRI, GAO, IMF, and other organizations.

This intermediate guide has three main sections:

Section 1, *Surveying the Model*, provides an introduction to the model's overall structure and components.

Section 2, *Installing the Model*, describes the installation procedure.

Section 3, *Running the Model*, references the documentation on various toolkits and outlines some advanced output analysis techniques.

Contents

1 Surveying the Model

SSASIM is a policy simulation model of the Old Age and Survivors Insurance (OASI) and the Disability Insurance (DI) programs. In addition to current-law policy, the model can simulate a variety of incremental reforms to these defined-benefit programs and a range of structural reforms that introduce a defined-contribution tier of self-managed individual accounts into social security.

A broad range of demographic and economic assumption parameters specify an environment. The environment can be represented as a *certain environment*, in which case it is simulated using one simulation *scenario*. Alternatively, an *uncertain environment* can be simulated by using Monte Carlo methods to sample many scenarios from dynamic probability distributions (i.e., stochastic processes) that represent the assumed expectations concerning the range of future values for the demographic and economic parameters.

A detailed set of specified policy parameters defines a *policy regime*, the implications of which are simulated in a *model run* that could assume either a certain or an uncertain environment. The consequences of a reform relative to current-law policy are determined by comparing the output results from two model runs that assume the same environment: one run that specifies the reform policy regime and another run that specifies the current-law policy regime. The model simulates the consequences of a policy regime for any specified number of years into the future. Notice that comparing the consequences of two different policy regimes in an uncertain environment involves comparing two probability distributions of each policy performance measure.

SSASIM actually consists of two models — a macro model and a micro model — that are integrated into a single computer program.

The *macro model* has two modes of operation: the default Cell-Based Actuarial (CBA) mode and the optional Over-Lapping Cohorts (OLC) mode.

The macro model's *CBA mode* represents the population with age-gender cells. It is quite similar to the model used by the Social Security Administration's Office of the Chief Actuary (OCACT) to produce the annual Trustees' Report (Trustees' Report 1999) and estimates of the program solvency consequences of reforms. When in CBA mode the macro model can be run in either an exogenous-growth mode or in an endogenous-growth mode. Only when growth is endogenous do the implications of the specified policy regime feed back through national savings and investment to influence the rate of economic growth using a neoclassical growth model.

The macro model's *OLC mode* uses the optional GEMINI add-on to SSASIM to simulate a small lifetime sample for each cohort born in 1935 and in each subsequent year to build up aggregate payroll tax revenues and OASDI benefit expenditures, which are then used to produce estimates of the same aggregate trust-fund statistics as estimated in the CBA mode.

The *micro model* has three modes of operation: the default Exemplary Cohort Individuals (ECI) mode, the optional Employer-Sponsored Pension (ESP) mode, and the optional Representative Cohort Sample (RCS) mode.

The micro model's *ECI mode* produces life histories for up to nine men and nine women from the same birth cohort. Because these cohort individuals may be paired into married couples, it is possible to calculate a full range of OASDI benefits as well as various benefit adequacy and contribution return measures for both the individuals and couples. The micro model is embedded inside the macro model so that the macro demographic and macro economic environment simulated in each scenario provides the context within which each individual's mortality, disability, employment, and account asset return history is simulated. In addition to being able to represent these collective risks that affect all individuals, the micro model represents individual risks caused by uncertainty regarding the age at which an individual will die and whether the individual will become disabled.

The micro model's *ESP mode* uses the optional PENSIM add-on to SSASIM to compute estimates of employer-sponsored pension benefits for each individual in a large sample of one or more specified birth cohorts.

The micro model's *RCS mode* uses the optional GEMINI add-on to SSASIM to compute the same kind of results as produced in the ECI mode, but for each individual in a large sample of one or more specified birth cohorts.

This first section of the guide describes the logical structure of SSASIM and provides details on the model's major features.

1.1 Model Architecture

The SSASIM model is designed to have a modular architecture. Modular architecture rejects the idea of the model as a single computer program that provides a large number of very different kinds of services: providing access to input data, calculating simulation results, and summarizing or visualizing output results. Instead, modular architecture calls for breaking up the model into a number of separate computer programs, each one of which specializes in providing one kind of service.

The main advantage of a model designed with a modular architecture is the increase in productivity that flows from the specialization allowed by the division of labor among component programs. Data-handling tasks can be performed by a program developed using a database management system that has superior data retrieval and ease-of-use features. Repetitive Monte Carlo calculations can be performed quickly by a compiled program that has been custom designed to do the necessary simulation computations. Statistical analysis and visualization of the simulated output can be done with easy-to-use spreadsheets, statistical packages, or custom-developed output analyzer programs.

This modular approach goes well beyond the recommendations of a National Academy of Sciences panel that reviewed the design of policy simulation models (Citro and Hanushek 1991, Cotton and Sadowsky 1991).

The modular architecture of the SSASIM model is shown in Figure 1. The model consists of two sets of disk files and three separate kinds of computer programs, all of which are accessed from the model's user interface program. The two sets of disk files are: the input database, and several output results files for each model run, which consists of the simulation of one policy regime under one set of assumptions. The three computer programs are: an user interface program, a stochastic simulator, and output analyzers.

1.1.1 User Interface Program

The RSF Toolkit provides the modern user interface to SSASIM and to the PENSIM and GEMINI add-ons. Read about the RSF Toolkit in the *Getting Started with the RSF Toolkit* ⟨http://www.polsim.com/rsfhelp.pdf⟩.

1.1.2 Input Database Tables

The model's input database is organized as a relational database. Relational databases facilitate the organization of complex data on the attributes of and relationships between several different kinds of objects in a way that is intuitive, efficient, and flexible. The utility of relational principles for database design and implementation is widely appreciated. For an accessible introduction to relational databases by one of the original developers, see Date (1983).

The input database tables are stored as a single-file SQLite database,

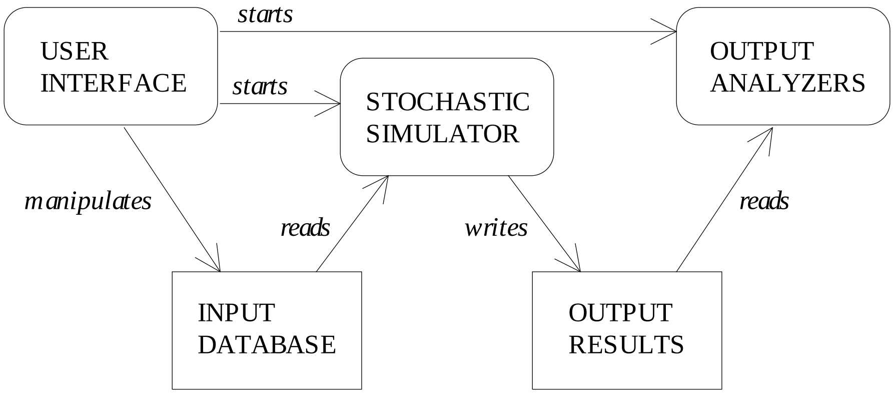

Figure 1: ***Modular Architecture of the SSASIM Model.*** *This architecture allows specialization among component programs and, if execution speed on a single computer chip is too slow, provides a development path to a multi-threaded architecture that permits distributed processing on multiple-chip computers.*

and the SSASIM installation contains a command-line tool (sqlite.exe) that can be used to manipulate the database in special cases where the standard methods are not adequate.

1.1.3 Stochastic Simulator Program

The model's stochastic simulator program reads model input parameters and historical demographic and economic data from the input database, performs the requested run, and writes detailed simulation output and summary results to output files. The simulator processes one or more scheduled model runs automatically in a non-interactive manner. Ill-specified model runs generate detailed error messages in an error-log file and are aborted, but do not prevent the processing of other scheduled runs. A complete transcript of a long series of runs can be saved to a run-log file rather than being shown on the screen.

The stochastic simulator program is written in the C++ language and is organized in a modular fashion using C++ classes that correspond to real-world entities related to social security. The simulator is written using object-oriented and other modern programming styles (Coplien 1992, for example).

1.1.4 Output Results Files

The SSASIM model's output results are written to text files that have been formatted to look like a spreadsheet. The output results files produced by the simulator during the course of each model run are labeled with the number of the run. This enables the storage of output results from many different runs and the comparison of results across runs long after the model runs were processed by the simulator. And because the specification of each run remains in the input database, the input parameters that generate the results for a run are completely documented.

The output results files can be read in any text editor, printed to paper, or processed by any application that imports tab-delimited text files such as a spreadsheet or statistical analysis program.

1.1.5 Output Analyzer Programs

The model's output analyzer programs include any software program that is appropriate for the desired post-simulation analysis of the output results. Because the output results are formatted in a manner that allows them to be easily imported into a wide variety of commercial software, there is enough flexibility in the selection of output analyzer programs to ensure that statistical analysis of the simulation output, output visualization, and preparation of presentation graphics are conducted with software that is both well suited to the task and familiar to the policy analyst using the model.

Although several output analyzer programs have been custom developed to facilitate routine analysis of output results, it should be noted that other options are available. Experience shows that one can never fully anticipate what post-simulation analysis must be done, and that unique, unexpected analysis needs arise routinely. The approach adopted here — writing detailed output and summary results from each model run to a set of text files, formatted in a way that allows them to be imported into a wide variety of commercial software — permits a flexible combination of anticipated analysis (using one of the existing output analyzer programs, for example) and unanticipated analysis (via custom-developed spreadsheets, for example).

1.2 Simulator's Macro Model of Program Finances

As mentioned above on page 1, the macro model can operate in two modes. This section discusses the macro model's CBA (cell-based actuarial) mode. The documentation of GEMINI, an optional add-on to SSASIM, contains a discussion of the OLC (over-lapping cohorts) mode (Holmer GEMINI Guide).

The macro model that is part of SSASIM is a cell-based dynamic policy simulation model that is logically similar to the actuarial model used by OCACT. While similar in many respects, it differs from that model in several important ways that are explained in this part of the guide.

Before discussing those major differences, it is important to describe briefly the nature of the logic in the macro model's population module. Mortality rates are not disease-specific as they are in the OCACT model. Each year the average probability of death is the same for everyone of the same gender and age, differs widely across age-gender groups, and declines at different rates across age-gender groups. This representation of mortality is based on that of Lee and Carter (1992). Fertility and immigration are represented in ways that are more similar to those used in the OCACT model.

1.2.1 Monte Carlo Methods for Stochastic Assumptions

The SSASIM macro model utilizes Monte Carlo simulation methods to derive the policy implications of uncertainty about the future values of at least thirteen demographic and economic assumption variables. This capability of representing collective risks implements recommendations by Citro and Hanushek (1997) and the Advisory Council (1997) to use stochastic simulation methods in place of the *ad hoc* high-cost, intermediate-cost, and low-cost projections of the Trustees' Report (1999).

For a classic introduction to Monte Carlo methods, see Hammersley and Handscomb (1964). The use of Monte Carlo methods requires the description of the range of future assumption variable values with probability distributions rather than point estimates. These probability distributions formalize expectations about the future value of the assumption parameters. These probability distributions take the form of stochastic processes in the SSASIM macro model because the assumption variables are time series.

Each model run samples from these probability distributions to generate the time path of assumption variables for each scenario. These randomly drawn assumptions are used to calculate for that scenario the consequences of that run's assumed policy regime. The model permits the specification of correlations among assumption variables (such that draws from the distributions are statistically related) and of feedback links (so that future values of variables are affected by prior values of other variables) so that complex interactions between social security operations, demographic variables, and economic variables can be represented. The results of this sampling for each scenario are written to the output results files.

1.2.2 Embedded Growth Model for Endogenous Growth

The SSASIM macro model has an option to use an embedded Solow-Swan neoclassical economic growth model with a Cobb-Douglas production function and labor-augmenting technological change (Barro and Sala-i-Martin 1995, Chapter 1). If this option is not selected, the macro model operates with exogenous economic growth as in the OCACT model.

The economic growth model is integrated into the broader macro model of social security in a manner similar that of (Aaron, Bosworth, and Burtless 1989). First, there are input links through which the broader model's variables affect the growth model's input variables. For example, the broader model's employment and productivity growth rate variables directly determine the growth model's corresponding input variables. Also, social security program finances (including individual account finances if specified in the policy regime) affect the growth model's national saving and investment rate. And second, there are output links through which the growth model's output variables feed back to influence variables in the broader model. For example, faster growth leading to higher output causes social security taxable earnings to be higher. Also, changes in the growth model's marginal productivity of capital causes changes in the broader model's asset returns, the basic values of which are determined by the stochastic processes described below beginning on page 14.

See Holmer (1999) for a detailed description of the embedded growth model, its input and output links to the broader model, and the baseline values assumed for the growth model's parameters.

1.2.3 High-Level Procedural Logic

The logic of the stochastic simulator's macro model is complex because it must represent both the internal dynamics of, and the interactions between, the population, the economy, and social security programs. The model relies on probability distributions that represent expectations concerning the major assumption variables, correlations between these assumption variables, feedback links between variables, and the internal procedural logic of the model, to represent population dynamics, economic development, and the workings of social security programs.

In the spirit of object-oriented programming, the simulator program is composed of discrete modules that correspond to real-world entities and perform functions related to those entities in the simulation process. Links between the modules are of two kinds. First, there are recursive links, in which current-year variable values produced by module are used in current-year calculations by another module. And second, there are lagged feedback links, in which the current-year variable values are used in the following year for calculations in another module. The use of lagged feedback links with a recursive module structure avoids the computational burden of simultaneous equation solution while allowing a reasonable characterization of general equilibrium interactions over the long time path of each simulation scenario.

Figure 2 presents a graphical representation of the structural relationships among the macro model's component modules, which are represented by boxes. Recursive links are represented with downward-pointing, solid-line arrows; lagged feedback links are represented by upward-pointing, broken-line arrows. The individual account module, which is not shown in this figure, is at the same level as the trust fund module, has the same recursive and lagged feedback links as the three social security modules.

After reading the input database for the run specification (starting variable values, expectations distributions, policy regime parameters, behavioral response coefficients, etc.), the stochastic simulator's macro model does the following for each multiple-year scenario in the run: First, the simulator randomly samples time paths for the assumption variables from their expectations distributions. Second, the simulator repeats a set of computations for each year in the scenario, in which the variables describing the status of the population, economy, and social security program at the end of the prior year are translated into values that describe the population, economy, and program status at the end of the current year. Third, the simulator saves results for that scenario for subsequent writing to the run's output results files.

Each simulation year's calculations involve a large number of recursive links among the simulator's modules. The selected recursive links shown in

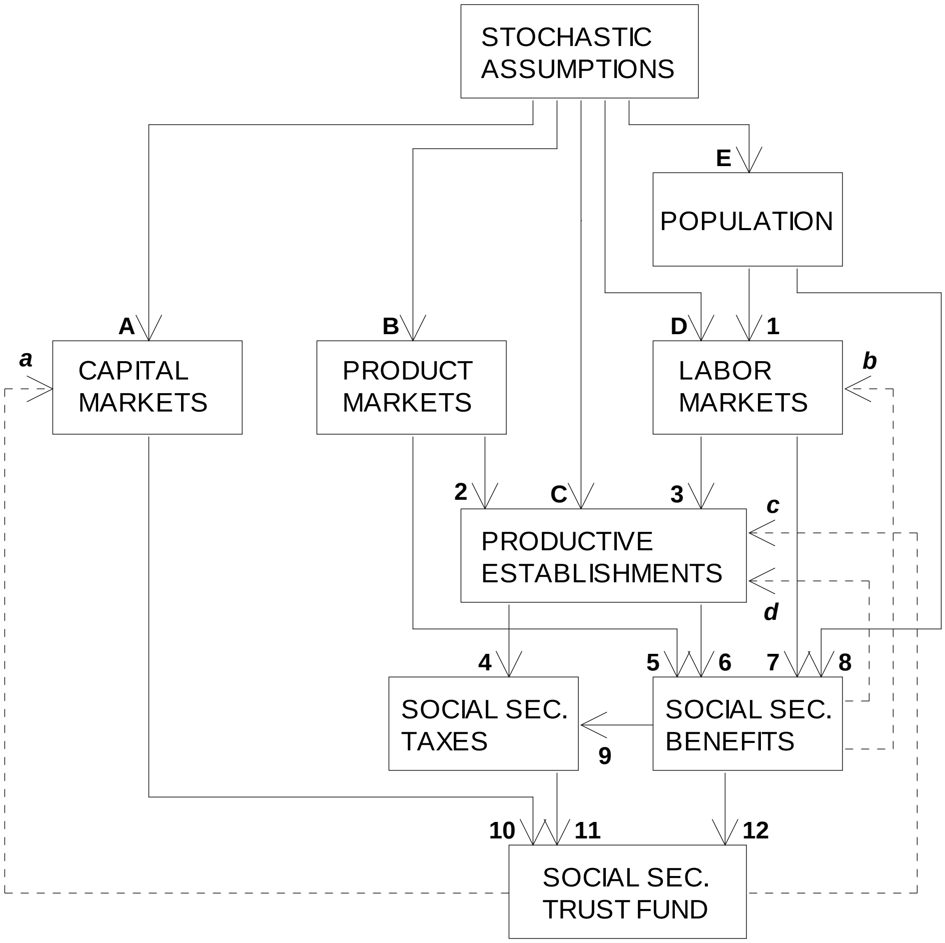

Figure 2: ***Modular Structure of the Macro Model in the Stochastic Simulator.*** *Individual modules (represented as boxes), selected recursive links (marked with numbers), and lagged feedback links (marked with lowercase letters) are described in the text. The individual account module, which is not shown in this figure, is at the same level as the trust fund module, has the same recursive and lagged feedback links as the three social security modules. The major stochastic assumption variables include: A, interest rate, corporate bond return, equity return; B, inflation; C, productivity growth, wage-share growth, hours-worked growth; D, female and male labor force participation, unemployment; E, fertility, immigration, mortality decline, disability incidence, disability recovery.*

Figure 2 are as follows: (1) Population in each age-gender cell is used to calculate employment in each cell. (2) Price levels are used to convert real values into nominal earnings and nominal output. (3) Aggregate employment is used to calculate aggregate real output and aggregate real earnings. (4) The level of aggregate earnings is used to calculate covered earnings and taxable payroll. (5) The rate of price inflation is used to calculate cost-of-living adjustments to social security benefits. (6) The level of nominal earnings is used to calculate social security indexed earnings for age-gender cells near retirement. (7) Employment in age-gender cells is used to calculate indexed earnings for age-gender cells near retirement. (8) Population in age-gender cells and the disability incidence and recovery factors are used to calculate the number of beneficiaries in each age-gender cell. (9) social security benefit amounts are used to calculate income tax revenue derived from taxing benefits. (10) Returns on different asset classes are used to calculate investment income to the social security trust funds. (11) Payroll and income tax revenue are used to calculate income to the trust funds. (12) Beneficiaries and their average benefit levels are used to calculate benefit and administrative expenses paid from the trust funds.

Lagged feedback links allow for the current-year values of variables to influence the subsequent-year values of other variables. The lagged feedback links shown in Figure 2 are as follows: (a) social security trust fund and/or individual account asset-allocation policies affect relative return spreads of equity to bonds (not yet implemented). (b) social security benefit policies affect retirement timing decisions (not fully tested). (c) social security trust fund and individual account finances affect (when economic growth is endogenous) national saving, investment, and output, which also affects the marginal productivity of capital and hence the level of asset returns in the capital markets. (d) social security benefit policies affect private retirement saving rates (not yet implemented).

Information about the internal logic of each of the simulator's modules is not presented here. A listing of all the simulator's component modules and their relationship to associated input database tables and output results files are presented below in Section 1.5.

1.2.4 Assumption Variable Dynamics

Now that the high-level procedural logic of the simulator's macro model has been outlined, the methods used to generate, for each stochastic scenario, the time paths for each assumption variable are explained. After explaining these methods, the discussion concludes by offering a Bayesian interpretation of the expectation distributions that are specified for each model run.

The stochastic simulator randomly generates values of the assumption variables for each scenario using the expectations specified for the run. The values for each assumption variable are a time series because the simulator's annual computations require a value for each assumption variable in each future year of the scenario. The structure of the multivariate stochastic process assumed to generate these sampled time series can accommodate a number of different methods of characterizing assumption dynamics, including the method used in the Trustees' Report (1999).

The Trustees' Report assumes a no-cyclical-fluctuations kind of time series for each assumption variable. Each variable is assumed to move linearly from its starting value to its ultimate value over the course of a fixed transition period. In every year following the transition period, the value of the variable is assumed to equal its ultimate value. This formulation implies that the whole time series can be describe by a single future value — the variable's assumed ultimate value.

Foster (1994) has used univariate autoregressive processes to represent the assumption dynamics of several economic variables that are important in the short-run current-law projection presented in the Trustees' Report. This kind of stochastic process is more realistic than the kind used in the Trustees' Report because it generates time series for the assumption variables that exhibit cyclical fluctuations.

The multivariate stochastic process that generates time series for the SSASIM macro assumption variables is similar to that used by Foster (1994), but it has been generalized in two important ways. First, the annual error terms of each stochastic variable can be correlated with other variable errors. And second, the stochastic process can represent uncertainty in the ultimate value for a variable.

The current version of the simulator has fifteen major assumption variables that are assumed to be stochastic: total fertility rate, net immigration flow, mortality decline rate, female labor force participation rate, male labor force participation rate, unemployment rate, inflation rate, productivity growth rate, wage share growth rate, hours worked growth rate, nominal interest rate (on Treasury bonds), disability incidence rate, disability recovery rate, equity return, and the rate spread between Treasury bills and Treasury bonds. The nature of the multivariate stochastic process used to generate the time paths of the fifteen major assumption variables is described below.

The simulator's assumptions module utilizes a very general kind of multivariate stochastic process that produces both cross-sectional correlation between the different variables and time-series fluctuations or cycles in the values of an individual variable. The value of an individual variable in a particular year is generated as the sum of a ultimate trend term and a cyclical deviation term.

A variable's ultimate trend term is assumed to start at its last observed historical value, move to its randomly generated ultimate value over a transition period following a linear trend (with an optional kink point), and stay at that ultimate value in all years following the transition period. The ultimate trend term values of the stochastic variables are generated for each scenario by drawing from a multivariate normal distribution, whose means, standard deviations, and correlation coefficients are specified to represent expectations about the uncertainty in the ultimate values for the stochastic assumption variables.

This formulation provides considerable flexibility in representing expectations about the long-run movement of the stochastic input variables. A certain historical trend (that is, starting and ultimate values equal to the historical value and zero variance for the ultimate value) can be specified, as well as trends that differ substantially from that experienced in the past.

A variable's deviation term is assumed to start at zero and move according to a vector autoregressive process with two lags — that is, a VAR(2) process — with a time-varying mean displacement. If v_t is a vector that represents the value of the stochastic variables for year t, then we have

$$v_t = u_t + d_t$$

where u_t denotes the ultimate trend term vector for year t and d_t the vector of cyclical deviations around the ultimate trend value for year t. The process that generates future values of the ultimate trend term has been described above. The stochastic process that generates the time path of deviations is as follows

$$d_t = \phi_1 d_{t-1} + \phi_2 d_{t-2} + \epsilon_t$$

where the distribution of the zero-mean, constant-variance error term vector ϵ, which is multivariate normal, and values of the elements of the ϕ_1 and ϕ_2 matrices must be specified.

Given a specification of the parameters and the assumption that $d_0 = d_{-1} = 0$, a time path of deviation terms can be generated from annual realizations of the independent error terms ϵ.

Results from the estimation of the parameters of a VAR(2) model for deviations in the inflation rate, nominal interest rate, equity return, and Treasury bill spread, using annual data from the late 1920s through to early 2000s, are presented below beginning on page 14.

The generality of this multivariate stochastic process implies that different kinds of time series generation methods can be specified as special cases.

Notice that specifying a zero variance for the error terms (ϵ) will produce zero deviation values for all future years. This special case corresponds to a stochastic process that generates values in accordance with the Trustees' Report method, and is the type of no-cyclical-fluctuations process employed to project population variables in Holmer (1996). Because the Trustees' Report method uses only one parameter (the ultimate value) to generate a time series for each assumption variable, the stochastic version of the Trustees' Report method can be interpreted from a Bayesian perspective. The multivariate normal distribution of ultimate values is essentially a Bayesian posterior distribution that combines prior beliefs with the implications of whatever data analysis has been conducted. Instead of assuming a point estimate of the ultimate value (as would a frequentist statistician), the stochastic version of the Trustees' Report method assumes a distribution for each ultimate value (as would a Bayesian statistician).

On the other hand, specifying a zero variance for the ultimate trend value and a constant positive variance for ϵ creates a special class of stochastic processes that are similar to those estimated by Foster (1994). Currently the simulator is structured to accept point estimates for the parameters of this VAR method of generating time series for the assumption variables. But there is no logical reason why the simulator could not be generalized to accept distributions for each parameter. This would permit a Bayesian implementation of the VAR method of generating assumption variable time series.

The specification of Bayesian versions of these different kinds of stochastic processes is desirable because the modeling uncertainty caused by the fact that model parameters are not known with certainty would not be ignored as it is when using point estimates for each model parameter. This advantage of the Bayesian approach to describing model parameters has been appreciated by economists (Geweke 1994, for example) as well as actuaries (Cairns 1994, for example). This approach would recognize all sources of uncertainty when estimating the future value of a simulated variable. On the other hand, Fair (1994, pp. 192 and 201–202) argues in favor of using the point estimate (not the distribution) of each parameter when using Monte Carlo methods to estimate the probability of a future event.

However conceptually desirable, this Bayesian approach has not been a practical alternative until recently. A set of new computational techniques have recently transformed the practice of estimating the distributions that describe each model parameter. While using these Markov chain Monte Carlo techniques to estimate Bayesian posterior distributions for the parameters of the model's stochastic process is beyond the scope of the current model, it appears as if it may be a desirable long-run goal for social security and other stochastic policy simulation models in the future.

1.2.5 Stochastic Dynamics of Asset Returns

This part of the guide presents results on the use of historical data to estimate the assumption parameters that determine the cyclical dynamics of four stochastic macroeconomic variables used in the model: the inflation rate, the nominal yield on Treasury bonds, the return on equities, and the yield spread between Treasury bills and Treasury bonds.

Cyclical fluctuations in these four macroeconomic variables are generated using a vector-autoregressive model with a two-year lag structure — a VAR(2) model — the parameters of which are estimated with annual historical data. In such a model, lagged values of all four variables plus an error term representing unpredictable shocks to the economy jointly determine the current value of each variable. This widely-used type of model guarantees that the interrelationships between the four variables in each randomly generated scenario will be historically plausible.

The macroeconomic VAR(2) model is estimated with data on deviations from sample averages, rather than their actual values, so that the estimated coefficients can be used in the simulation model and, at the same time, provide the option of setting the future mean value of each macroeconomic variable at a level different from its historical average.

If π_t the deviation in the decimal CPI inflation rate, i_t denotes the deviation in the natural logarithm of the decimal nominal Treasury bond yield (that is, the nominal interest rate) for year t, e_t the deviation in the log equity (gross) return, s_t denotes the difference is the log decimal Treasury bill yield and the log decimal Treasury bond yield, and $M_t = (\pi_t, i_t, e_t, s_t)'$ denotes a column vector of the four deviation variables, then the VAR model can be expressed as follows

$$M_t = \Psi_1 M_{t-1} + \Psi_2 M_{t-2} + \epsilon_t$$

where ϵ_t denotes a column vector of error terms that have a zero-mean multivariate normal distribution and Ψ_1 and Ψ_2 are 4x4 matrices of estimated coefficients. Note that this specification allows contemporaneous correlation among the error terms for the four variables, enabling the model to simulate historically plausible macroeconomic shocks. The parameters estimated for this VAR(2) model are used by the simulation model to generate fluctuations in these variables that will cause, through the simulation logic, both fluctuations in program benefits and in asset returns.

The annual historical data from 1954 through 2009 that are used to estimate this VAR(2) model are obtained from two sources. The inflation rate, equity return on large company stocks, and Treasury bill return are drawn from *Ibbotson SBBI 2010 Classic Yearbook: Market Results for Stocks, Bonds, Bills and Inflation, 1926–2009*, Chicago: Morningstar, 2010. And the nominal Treasury bond yield is drawn from the web site of the SSA Office of the Chief Actuary.

Tables 1–4 on pages 16–17 contain the estimated structural parameters Ψ_1 and Ψ_2 of this 4x4 VAR(2) model.

The matrix of estimated correlation coefficients of this VAR(2) model's error terms ϵ is shown in Table 5.

The estimation results for the deviation in equity returns suggests that returns are generated by a lognormal random walk process. However, in recent years empirical evidence has mounted in favor of a mean-reversion process involving the equity dividend-price ratio as described by Campbell, Lo, and MacKinlay (1997, p. 271) and Cochrane (1997, pp. 8–14).

SSASIM has a simple way of approximating this mean-reversion behavior in equity returns. The log equity return simulated as the sum of the assumed long-run mean log equity return and the deviation generated by the VAR(2) model described on page 15 has a mean-reversion term added to it. If the long-run mean is denoted by M, then the mean-reversion term for year t is denoted by R_t and is defined as

$$R_t = \rho(M - A_t)$$

Table 1: **VAR(2) Model's Inflation Coefficients in Ψ_1 & Ψ_2.** The zero coefficients were imposed when the unrestricted estimation produced parameters neither of which were significant at the 0.15 level.

Lagged Variable	Coefficient	Std Error	t-Statistic
Inflation(–1)	0.902	0.134	6.71
Inflation(–2)	–0.188	0.135	–1.39
Tbond(–1)	0.0	—	—
Tbond(–2)	0.0	—	—
Equity(–1)	0.0	—	—
Equity(–2)	0.0	—	—
Tbill(–1)	0.0	—	—
Tbill(–2)	0.0	—	—
Standard Error σ_ϵ	0.018883		
R-squared statistic	0.591		
Durbin-Watson statistic	1.896		

Table 2: **VAR(2) Model's Tbond Coefficients in Ψ_1 & Ψ_2.** The zero coefficients were imposed when the unrestricted estimation produced parameters neither of which were significant at the 0.15 level.

Lagged Variable	Coefficient	Std Error	t-Statistic
Inflation(–1)	2.398	1.188	2.02
Inflation(–2)	–1.172	1.145	–1.02
Tbond(–1)	0.671	0.160	4.21
Tbond(–2)	0.221	0.149	1.49
Equity(–1)	0.0	—	—
Equity(–2)	0.0	—	—
Tbill(–1)	0.0	—	—
Tbill(–2)	0.0	—	—
Standard Error σ_ϵ	0.1412		
R-squared statistic	0.906		
Durbin-Watson statistic	1.937		

Table 3: **VAR(2) Model's Equity Coefficients in Ψ_1 & Ψ_2.** The zero coefficients were imposed when the unrestricted estimation produced parameters neither of which were significant at the 0.15 level. Because all the structural coefficients were statistically insignificant, a constant term was added to this, and only this, equation, and the estimated value of the constant term is essentially zero: –4.7e-9.

Lagged Variable	Coefficient	Std Error	t-Statistic
Inflation(–1)	0.0	—	—
Inflation(–2)	0.0	—	—
Tbond(–1)	0.0	—	—
Tbond(–2)	0.0	—	—
Equity(–1)	0.0	—	—
Equity(–2)	0.0	—	—
Tbill(–1)	0.0	—	—
Tbill(–2)	0.0	—	—
Standard Error σ_ϵ	0.174		
R-squared statistic	0.000		
Durbin-Watson statistic	1.993		

Table 4: **VAR(2) Model's Tbill Coefficients in Ψ_1 & Ψ_2.** The zero coefficients were imposed when the unrestricted estimation produced parameters neither of which were significant at the 0.15 level.

Lagged Variable	Coefficient	Std Error	t-Statistic
Inflation(–1)	3.693	2.614	1.41
Inflation(–2)	–4.653	2.466	–1.89
Tbond(–1)	1.744	0.320	5.44
Tbond(–2)	–1.513	0.293	–5.17
Equity(–1)	1.675	0.224	7.48
Equity(–2)	–0.137	0.273	–0.50
Tbill(–1)	0.885	0.143	6.18
Tbill(–2)	–0.512	0.137	–3.73
Standard error σ_ϵ	0.2662		
R-squared statistic	0.786		
Durbin-Watson statistic	1.886		

Table 5: **VAR(2) Model's Correlation Coefficient Matrix for ϵ.**

Variable	Inflation	Tbond	Equity	Tbill
Inflation	1.000			
Tbond	0.513	1.000		
Equity	–0.233	0.038	1.000	
Tbill	–0.056	–0.123	–0.153	1.000

where A_t notes the average of log equity returns in the N years prior to year t. The best fit to Ibbotson data on equity returns for 1933–2005 is when $N = 5$; in this case, the estimated value of ρ is 0.428 with the coefficient being significantly different from zero at the 0.087 level.

In practice, SSASIM can be used to simulate equity returns either with or without mean reversion. With mean-reversion, the above values of N and ρ are used along with the estimated parameters of the VAR(2) deviation process described above, except that the standard deviation of the log equity return deviation is adjusted so that the volatility of equity returns matches some desired level (for example, 20.2 percent as reported in Ibbotson for the 1926–2005 period, or 20.5 percent as reported by Ibbotson for the 1926–2009 period). Likewise, when no mean reversion is assumed, ρ is assumed to be zero and the VAR(2) standard deviation of log equity return deviations is adjusted upward (to roughly 0.19) so that the simulated volatility is at the desired level.

This discussion has little to do with the question of the level of the long-run mean equity return in the future. For an extended discussion of issues relevant to projecting the level of the assumed long-run mean equity return, see Diamond (1999), who uses the "Gordon growth model" to organize the analysis of these issues. In particular, he discusses the inconsistency of pairing the intermediate-cost economic projection assumptions with the historical mean equity return, unless there is a sharp fall in equity prices in the next decade or so.

1.2.6 Markov Dynamics of Beneficiaries

In the SSASIM macro model operating in CBA mode, the number of people moving on to and off of the OASI and DI programs is controlled by Markov transition rates that are age and gender specific. These Markov transition rates can be specified to vary over time, representing changes in behavior either caused by policy reforms or arising from other social developments. These transition rates are grouped into age-gender matrices for increments and for decrements to the stock of program beneficiaries. The transition rates are defined over a Markov process in which people move between three states: (1) being an OASI beneficiary, (2) being a DI beneficiary, and (3) being neither an OASI nor a DI beneficiary.

This approach is similar to that used in the OCACT model to simulate the number of new DI beneficiaries and the number of DI beneficiaries who leave the program for reasons other than death. This so-called increment-decrement-rate approach is not used in the OCACT model for the OASI program, which is instead modeled using a so-called prevalence-rate approach that specifies the fraction of people who are beneficiaries at each age.

In the SSASIM macro model operating in CBA mode, the average initial benefit for new beneficiaries in each age-gender cell is calculated as the product of two terms: the economy-wide average earnings level for that year and the relative benefit ratio (of average initial benefits to economy-wide average earnings) for that age-gender cell. This means that reforms that affect initial benefit levels need to be characterized in the macro model in terms of their overall effect on initial benefit levels, the size of which effect may vary in future years.

This reduced-form approach is required because the cell-based macro model does not have longitudinal earnings histories with which to compute initial benefits using the structural policy parameters in the AIME, PIA, and benefit formulas. On the other hand, the SSASIM micro model (see below) does simulate such earnings histories, and therefore, utilizes this structural approach to initial benefit calculation.

1.2.7 Policy Performance Measures of Program Finances

The SSASIM macro model produces for each simulation scenario numerous measures concerning the program finances of the defined-benefit tier of both the OASI and DI programs, and for the combined OASDI program. Among these measures are the 75-year actuarial balance (which is generalized so that the length of the actuarial computation period can be shorter or longer than 75 years), the trust fund exhaustion year, and the summarized cost rate used to compute the actuarial balance. For technical definitions of the actuarial balance, the summarized cost rate, and the summarized income rate, which is simply the sum of the actuarial balance and the summarized cost rate, see the discussion in the Trustees' Report (1999, pp. 21–24).

The variability of program solvency measures across simulation scenarios shows the sensitivity of program finances to changes in assumed demographic and economic assumptions. This use of Monte Carlo methods to estimate the sensitivity is superior to the method of comparing results from a couple of *ad hoc* alternative sets of non-stochastic assumptions.

A complete list of the policy performance measures of program finances is available in the online help sections concerning the OUTPUT table.

1.3 Simulator's Micro Model of Individual Lives

As mentioned above on page 2, the micro model can operate in three modes. This section discusses the micro model's ECI (exemplary cohort individuals) mode.

The documentation for PENSIM, an optional add-on to SSASIM, contains a discussion of the ESP (employer-sponsored pension) mode (Holmer et al. PENSIM Overview).

The documentation for GEMINI, an optional add-on to SSASIM, contains a discussion of the RCS (representative cohort sample) mode (Holmer GEMINI Guide).

The micro model that is part of SSASIM has logic similar to a full-blown dynamic simulation model, but operates on a small sample limited to nine men and nine women all of whom are born in the same year (1935 or afterward). The birth year of these cohort individuals can differ in otherwise identically specified model runs, permitting analysis of how a policy reform would affect individuals in several different birth cohorts. The model simulates a complete earnings history as well as disability and mortality events, and therefore, can estimate using structural policy parameters the annual contributions and benefits received by each individual from both the defined-benefit and defined-contribution tiers of the OASI and DI programs. The model allows individuals to be considered as a couple who have children at specified ages, so the full range of retirement, survivors, and disability benefits can be simulated.

The SSASIM micro model is best viewed as a generalization of the stylized individuals often used in social security policy analysis to calculate contribution return and benefit adequacy measures for low, average, and high lifetime earners Steuerle and Bakija (1994, for example). The micro model provides more flexibility than traditional analysis of this kind because it combines the use of Monte Carlo methods to represent risks facing the individuals (e.g., uncertain asset returns). The recognition of these risks is accomplished because the micro model of individual lives is embedded inside the macro model that simulates the aggregate economic environment within which the cohort individuals' lives play out.

The use of Monte Carlo methods to represent risks in the micro model, the high-level logic of the micro model, and the kinds of contribution return and benefit adequacy policy performance measures produced by the micro model are discussed in more detail below.

1.3.1 Monte Carlo Methods for Uncertain Lives

Depending on the specified input parameters, the individual life histories generated by the micro model may be as stylized as those used in conventional social security analysis, or they may be more realistic exhibiting career earnings growth, inflation, uncertain asset returns, periods of non-employment, and disability.

The stylized individual life histories are produced by the micro model operating in Exemplary Cohort Individual (ECI) mode. In this mode, economic variables and asset returns are uncertain across scenarios. When the micro model operates in either the Employer-Sponsored Pension (ESP) mode or the Representative Cohort Sample (RCS) mode, the full range of uncertainty is present creating variation in life histories both within and across macro scenarios. The simulated lifetime earnings histories have the full variety found in recent empirical studies (Bosworth, Burtless, and Steuerle 1999).

1.3.2 High-Level Procedural Logic

The high-level logic of the SSASIM micro model parallels that of the macro model shown in Figure 2. During each year, after the calculations are completed in a macro module, the corresponding micro model calculations are performed.

The bulk of the SSASIM micro model's logic involves computing OASI and DI benefits given the life histories of the individual and spouse, and given the value of the structural benefit policy parameters specified for each year of the model run. These structural benefit policy parameters permit the specification of a wide range of benefit policy reforms to the AIME, PIA, individual benefit, maximum family benefit, and earnings test formulas.

1.3.3 Policy Performance Measures of Contribution Return

The SSASIM micro model produces for each simulation scenario numerous measures concerning the return on contributions to a two-tier social security program. These measures, which are often called money's worth measures, include present value of lifetime benefits (PVB), present value of lifetime contributions and taxes (PVT), net present value (PVB-PVT), internal rate of return (the discount rate that makes PVB equal PVT), and the payback ratio (PVB/PVT). These contribution return measures are computed and written to separate output results files for each simulated person individually and for each couple collectively. They are calculated for the OASI and DI programs separately. A value of each measure is calculated for each simulation scenario, which permits an analysis of the riskiness of contribution returns under different policy regimes.

A complete list of the policy performance measures of contribution return is available in the online help sections concerning the OUTPUT table.

1.3.4 Policy Performance Measures of Benefit Adequacy

The SSASIM micro model produces for each simulation scenario numerous measures concerning the adequacy of benefits paid by a two-tier social security program. These measures include average real benefit level (averaged over all retirement years), replacement rate, and low-benefit avoidance rate (the fraction of retirement years in which the benefit exceeds some low-benefit benefit threshold specified by the policy analyst using the model). Analogous measures are calculated for survivors benefits received before retirement, and for disability benefits. A value of each measure is calculated for each simulation scenario, which permits an analysis of the riskiness of benefit adequacy under different policy regimes.

A complete list of the policy performance measures of benefit adequacy is available in the online help sections concerning the OUTPUT table.

1.4 Relationship Between Macro and Micro Models

In SSASIM the micro model of cohort individuals is enclosed inside the macro model of program finances. This nature of this enclosure is such that all the assumptions used by the macro model and all simulation results produced by the macro model are visible to the micro model. But the assumptions specified for use by the micro model are not visible in the macro model. This one-way information sharing relationship allows the simulated macro variables to define the environment within which the lives of the cohort individuals play out; it also implies that the structural benefit policy parameters used to calculate defined benefits in the micro model have no effect on the operation of the macro model.

As explained in the detailed discussion of the input database tables contained in Section 1.5), the COHORT input database table and all its child tables contain assumption and policy parameters that are used only in the micro model of cohort individuals. All other input database tables contain assumption and policy parameters for the macro model.

1.5 Model's Structure in Detail

The final part of this first section of the guide identifies all the input database tables, all the simulator logic modules, and all the output results files associated with SSASIM. Brief descriptions of each table, module, and file are given. Following those descriptions, the relationships between the tables, modules, and files are shown in Table 6 that begins on page 41. This table provides a more complete enumeration of the model's modules than is contained in Figure 2.

It is important to note that the COHORT input database table and all its child tables contain assumption and policy parameters that are used only in the micro model of cohort individuals. All other input database tables contain assumption and policy parameters for the macro model of program finances. Assumptions and policy parameters for the macro model are visible to and used by the micro model when appropriate, but assumptions and policy parameters for the micro model are invisible to, and therefore, not used in the macro model.

1.5.1 Input Database Tables

The model's stochastic simulator program reads over eighty input database tables in order to specify a model run. Each table contains an id number field — called a key in relational database terminology — that identifies table rows (or records, as the table rows are sometimes called). The tables are structured in a hierarchy in which the relevant row or group of rows in a child table is identified by the value of a table-specific row id number — called a foreign key — in the parent table. For more information about the logical structure of a relational database, see Date (1983).

Details on each table field can be found using the documentation that can be accessed from the RSF Toolkit.

The names of the input database tables are presented below in the order that they are displayed in the documentation with the degree of indention representing their hierarchical level:

- **QUEUE** Queue table rows contain the id numbers of runs that are scheduled to be processed by the stochastic simulator.
  - **RUN** Run table rows contain the starting year of the simulation, number of years in each scenario, and the number of scenarios, as well as the id numbers of the run table's associated row(s) in the following child tables.
    - **OUTPUT** Output table rows contain information about which output results files are selected for tabulation and writing.
    - **ASSUMP** Assumption table rows contain id numbers of the assumption table's associated row(s) in the following child tables. For details on assumption variable dynamics, see Section 1.2.4.
      - **TRND_MV** Trend mean variance table rows contain information about the mean and standard deviation of the ultimate trend value for each of the thirteen major assumption variables.
      - **TRND_CC** Trend correlation coefficient table rows contain correlation coefficients among the ultimate trend values.

- **DEV_CC** Deviation correlation coefficient table rows contain correlation coefficients among the error terms in the VAR(2) deviation process.
- **DEV_P** Deviation VAR coefficients table rows contain coefficients for the VAR(2) deviation process.

POP Population table rows contain initial-year values and transition information for the three population assumption variables, as well as the id numbers of the population table's associated row(s) in the following child tables.

- **POP_DEL** Annual offset values for each of the three population assumption variables.
- **POP_GA0** Gender-age initial-year population table rows contain the number of people in each gender-age cell in the simulation's initial year.
- **POPDYN** Population dynamics table rows contain id numbers of the population dynamics table's associated row(s) in the following child tables.
  - **FR_A** Age-specific relative fertility rate table rows contain age-specific total fertility rates that define the age pattern of fertility for the specified calendar year.
  - **IMM_GA** Gender-age relative net immigration table rows contain net immigration flows in each gender-age cell that define the pattern of net immigration for the specified calendar year.
  - **MDR_GA** Gender-age relative mortality decline rate table
    rows contain rates of decline in mortality rates for each gender-age cell that are used to determine rates of mortality decline relative to the population-wide average rate for the specified calendar year.
  - **SP_PROB** Gender-age spouse probability table rows contain the fraction of each gender-age cell that is assumed to have a living spouse for the specified calendar year.
  - **SP_AGED** Gender-age spouse age difference table rows contain the average age difference between married peo-

ple in that gender-age cell and their spouse for the specified calendar year.

LABMRKT Labor market table rows contain initial-year values and transition information for the three labor market assumption variables, as well as the id numbers of the labor market table's associated row(s) in the following child tables.

- **LABDYN** Labor market dynamics table rows contain id numbers of the labor market dynamics table's associated row(s) in the following child tables.
  - **LFPR_GA** Gender-age relative labor force participation rate table rows contain participation rates for each gender-age cell that are used to determine participation rates relative to the gender-wide average rate.
  - **UNR_GA** Gender-age relative unemployment rate table rows contain unemployment rates for each gender-age cell that are used to determine unemployment rates relative to the labor-force-wide average.
  - **EARN_GA** Gender-age relative earnings table rows contain earnings ratios that specify the assumed ratio between that gender-age cell's average earnings and all-worker average earnings.

PRDMRKT Product market table rows contain the initial-year value and transition information for the inflation rate assumption variable, as well as information about the relationship between CPI and GDP-deflator measures of inflation.

ESTAB Establishment table rows contain initial-year values and transition information for the three productivity and output related assumption variables, as well as assumed parameter values for the embedded economic growth model and its links to the broader model.

CAPMRKT Capital market table rows contain the initial-year value and transition information for asset returns.

BEN_DI DI benefit table rows contain initial-year values and transition information for the two disability assumption variables, as well as information about program administrative costs and the id numbers of the DI benefit table's associated row(s) in the following child tables.

- **BEND_DI** DI benefit policy parameters that have a time dimension.
- **NBEN_DI** Gender-age initial-year DI number of beneficiaries table rows contain the number of DI beneficiaries in each gender-age cell for the initial year.
- **ABEN_DI** Gender-age initial-year average DI benefit table rows contain the average benefit of DI beneficiaries in each gender-age cell for the initial year.
- **BDYN_DI** DI benefit policy dynamics table rows contain information about how DI benefit policy changes over the course of the simulation years, which is in the form of id numbers of the DI benefit policy dynamics table's associated row(s) in the following child tables.
  - **RBEN_DI** Gender-age relative DI benefit table rows contain average DI benefits for new beneficiaries in each gender-age cell expressed relative to earnings.
  - **INCR_DI** Gender-age relative DI disability incidence rate table rows contain incidence rates (or Markov transition probabilities on to DI) for each gender-age cell that are used to determine incidence rates relative to the population-wide average DI incidence rate.
  - **RMRT_DI** Gender-age relative DI mortality rate table rows contain relative mortality factors that determine the mortality rate of DI beneficiaries in each gender-age cell relative to the mortality rate of the non-DI population in each cell.
  - **RECR_DI** Gender-age relative DI recovery rate table rows contain recovery rates (or Markov transition probabilities off of DI for reasons other than death) for each gender-age cell that are used to determine recovery rates relative to the beneficiary-wide average DI recovery rate.

BEN_RI OASI benefit table rows contain information about program administrative costs and the id numbers of the OASI benefit table's associated row(s) in the following child tables.

- **BEND_RI** OASI benefit policy parameters that have a time dimension.

NBEN_RI Gender-age initial-year OASI number of beneficiaries table rows contain the number of OASI beneficiaries in each gender-age cell for the initial year.

ABEN_RI Gender-age initial-year average OASI benefit table rows contain the average benefit of OASI beneficiaries in each gender-age cell for the initial year.

BDYN_RI OASI benefit policy dynamics table rows contain information about how OASI benefit policy changes over the course of the simulation years, which is in the form of id numbers of the OASI benefit policy dynamics table's associated row(s) in the following child tables.

RBEN_RI Gender-age relative OASI benefit table rows contain average OASI benefits for new beneficiaries in each gender-age cell expressed relative to earnings.

INCR_RI Gender-age OASI incidence rate table rows contain incidence rates (or Markov transition probabilities on to OASI) for each gender-age cell.

DECR_RI Gender-age relative OASI decrement rate table rows contain decrement rates (or Markov transition probabilities off of OASI for reasons other than death) for each gender-age cell.

TAXEARN Aggregate taxable earnings table rows contain parameters used to convert total earnings into payroll taxable earnings.

COVERED Covered table rows contain parameter used to specify fraction of government-sector job starts that are not covered by OASDI.

MTE Maximum taxable earnings table rows contain parameters used to specify reform induced changes in the index level of maximum taxable earnings.

TAX_DI DI tax table rows contain parameters concerning the income taxation of benefits, as well as id numbers of the DI tax table's associated row(s) in the following child table.

TAXR_DI DI rate table rows contain scheduled payroll tax rates and general revenue transfer rates by simulation year.

TAX_RI OASI tax table rows contain parameters concerning the income taxation of benefits, as well as id numbers of the OASI tax table's associated row(s) in the following child table.

- **TAXR_RI** OASI rate table rows contain scheduled payroll tax rates and general revenue transfer rates by simulation year.

FUND_DI DI trust fund table rows contain information about the initial year's investments in corporate bonds and equity, and parameters that characterize the fund's asset management strategy, as well as id numbers of the DI trust fund table's associated row(s) in the following child tables.

- **IPAA_DI** DI trust fund investment-portfolio asset allocation fractions table rows contain asset-allocation shares for all simulation years.
- **STB0_DI** DI trust fund initial-year special Treasury bond holdings table rows contain coupon rate, maturity, and principal amount of all special-issue Treasury issues held by the DI fund in the initial simulation year.

FUND_RI OASI trust fund table rows contain information about the initial year's investments in corporate bonds and equity, and parameters that characterize the fund's asset management strategy, as well as id numbers of the OASI trust fund table's associated row(s) in the following child tables.

- **IPAA_RI** OASI trust fund investment-portfolio asset allocation fractions table rows contain asset-allocation shares for all simulation years.
- **STB0_RI** OASI trust fund initial-year special Treasury bond holdings table rows contain coupon rate, maturity, and principal amount of all special-issue Treasury issues held by the OASI fund in the initial simulation year.

FUND_RD OASDI trust fund table rows contain information about the initial year's investments in corporate bonds and equity, and parameters that characterize the fund's asset management strategy, as well as id numbers of the OASDI trust fund table's associated row(s) in the following child tables.

- **IPAA_RD** OASDI trust fund investment-portfolio asset allocation fractions table rows contain asset-allocation shares for all simulation years.
- **STB0_RD** OASDI trust fund initial-year special Treasury bond holdings table rows contain coupon rate, maturity, and principal amount of all special-issue Treasury issues held by the OASDI fund in the initial simulation year.

ACCT Individual account table rows contain information about the timing of initial account withdrawal, whether any annuity withdrawal is in the form of an inflation-indexed annuity, and if so, information about how that indexed annuity would be priced, as well as id numbers of the account table's associated row(s) in the following child tables.

- **ACCTBAL** Gender-age initial-year account balance table rows contain initial-year aggregate account balances for each gender-age cell.
- **ACCTAPY** Gender-age initial-year account annuity payments table rows contain initial-year aggregate annuity payment amounts for each gender-age cell.
- **ACCTDYN** Account policy dynamics table rows contain information about how account policy and behavior changes over the course of the simulation years, which is in the form of parameters (like the contribution rate) and in the form of id numbers of the account dynamics table's associated row(s) in the following child table.
- **ACCTAA** Gender-age account asset allocation table rows contain asset-allocation shares for each gender-age cell over the course of all the years in a simulation scenario.

OLC Contains information about the many cohort samples produced by the optional GEMINI add-on that are required when the SSASIM macro model operates in OLC mode.

ESP Contains execution information about the pre-specified runs of the optional PENSIM add-on that are to be executed when the SSASIM micro model operates in ESP mode.

RCS Contains execution information about the pre-specified runs of the optional GEMINI add-on that are to be executed when the SSASIM micro model operates in RCS mode.

- **COHORT** Cohort table rows contain the birth year of the specified cohort, as well as the id numbers of the cohort table's associated row(s) in the following child tables.
  - **POLICY** Cohort policy table rows contain the id numbers of associated row(s) in the following child tables.
    - **INSURED** This policy table contains dynamic quarters of coverage policy parameters by year.
    - **AIME** This policy table contains dynamic wage indexing policy parameters by year.
    - **PIA** This policy table contains just static PIA calculation policy parameters.
      - **DIVTIME** This policy table contains dynamic divorce-timing policy parameters by year for OASI.
      - **PIADYNR** This policy table contains dynamic PIA calculation policy parameters by year for the OAI program.
      - **PIADYNS** This policy table contains dynamic PIA calculation policy parameters by year for the SI program.
      - **PIADYND** This policy table contains dynamic PIA calculation policy parameters by year for the DI program.
      - **WEP_GPO** This policy table contains dynamic WEP-GPO benefit reduction policy parameters.
    - **MINPIA** This policy table contains flat or minimum PIA policy parameters.
    - **MFB** This policy table contains just static maximum family benefit policy parameters.
      - **MFBDYN** This policy table contains just dynamic maximum family benefit policy parameters by year.
    - **ETEST** This policy table contains dynamic earnings test policy parameters by year.
    - **OAIBEN** This policy table contains dynamic spouse and retired-worker benefit policy parameters by birth year.
    - **SIBEN** Table rows contain the id numbers of associated row(s) in the following child tables.

WWBEN This policy table contains dynamic aged widow(er) survivor benefit policy parameters by birth year.

WPBEN This policy table contains dynamic widowed parent survivor benefit policy parameters by year.

DWBEN This policy table contains dynamic disabled widow(er) survivor benefit policy parameters by birth year.

DIBEN This policy table contains dynamic disability benefit policy parameters by year.

INDACCT This policy table contains dynamic individual account policy parameters by year.

ACCTBEH Account behavior table rows contain individual account behavior assumptions.

INDS Cohort individuals table rows contain the id numbers of the cohort individual table's associated row(s) in the following child table.

IND Cohort individual table rows contain information on the aggregation weight, gender and assumed retirement age, as well as the id numbers of the IND table's associated row(s) in the following child tables.

EARNPCT Individual relative earnings table rows contain information on annual earnings over the individual's career measured relative to the national average wage index (AWI) for that year.

HISTORY History table rows contain information on various economic and program variables in years before the simulation's initial year.

HISTMR Mortality rates by age and gender for various year up to and including the simulation's initial year.

HISTMDR Mortality decline rate history table rows contain overall female and male mortality decline rates for years before the simulation's initial year. There is a set of rows corresponding to each starting-year's mortality rates, with each set producing historically accurate backcasted life expectancies for those born in 1930–1990.

1.5.2 Simulator Logic Modules

The model's stochastic simulator is composed of more than thirty modules or C++ classes. The source code for the each of the modules is spread across two files: a .h file, which contains the public interface to the module and defines the class's private data structures, and a .cpp file, which contains the module's private logic and algorithms.

If you have requested the source-code documentation, details about the field contents of tables associated with a module can be obtained by reading each table's field names and definitions, which are located toward the bottom of the .h file. The .cpp file's read function contains the logic involved in reading the tables and placing the read data into module data structures, which are defined in the private section of the .h file. Selectively reading the code in this order will provide a detailed understanding of each input database table and how the information read from a table contributes to defining a model run.

Details about the output results files associated with a module can be obtained by reading each results file's field names, which are located in out_EXT.h files (where EXT is the output results file's three-letter extension). The .cpp file contains a result_field_values function that assigns to each result field simulated values, which are defined in the private section of the .h file. Selectively reading the source code in this order will provide a detailed understanding of each output results file and how the information written to a file is calculated by the module.

Most of the modules are relatively independent, but some are related to other modules in one of two ways. The first kind of relationship between two modules is the "used by" relationship. The second kind is the "derived from" relationship. One module can use another just as a carpenter can use a hammer. The derivation relationship is one in which a particular entity is a special case of a general category. For example, a square can be said to be derived from the more general rectangle entity because it inherits all the general characteristics of the rectangle. For more on C++ derivation and inheritance hierarchies in object-oriented programming, see Coplien (1992), for example.

The names of the program modules are presented in the order that they are called during scenario calculations by the simulator's run module with indention indicating either used by or derivation from the higher level module:

run Run module has capabilities of reading input database tables, conducting the simulation calculations (by calling the following modules in roughly the order they are listed), and writing the specified output results files. The run module's calculate_results function contains the high-level procedural logic of the simulation calculations, including the sequence of each scenario year's calculations.

output Output module knows which output result files have been specified for this model run.

assump Assumption module generates the future time paths of thirteen major assumption variables using the appropriate version of the assumption stochastic process.

pop Population module advances the population from year to year using the time paths of fertility, immigration, and mortality decline generated by the assumption module.

labmrkt Labor market module advances the employment from year to year using the time paths of female and male labor force participation and unemployment rates generated by the assumption module.

prdmrkt Product market module advances the CPI index and GDP deflator from year to year using the time path of the inflation rate generated by the assumption module.

estab Establishment module advances GDP and total labor earnings from year to year using three output-related variables generated by the assumption module. These calculations can be exogenous or endogenous if the embedded economic growth model and its links to the broader model are activated.

capmrkt Capital market module generates the time paths of three asset returns using the specified asset returns equations.

ben Benefit module calculates the benefit cost rate for each simulation year.

> **ben_di** DI benefit module is derived from benefit module to calculate the benefit cost rate via a Markov process of beneficiary movement on and off the program and differential handling of new and existing benefit levels.

- **ben_ri** OASI benefit module is derived from benefit module to calculate the benefit cost rate via a Markov process of beneficiary movement on and off the program and differential handling of new and existing benefit levels.
- **ben_rd** OASDI benefit module is derived from benefit module and calculates several OASDI benefit variables as the sum of corresponding OASI and DI benefit variables.

tax Tax module calculates aggregate and maximum OASDI taxable earnings and program-specific tax revenue from both the payroll tax and the income tax on social security benefits using simulation results.

- **tax_di** DI tax module is derived from the tax module to perform the DI tax calculations.
- **tax_ri** OASI tax module is derived from the tax module to perform the OASI tax calculations.
- **tax_rd** OASDI tax module is derived from the tax module and calculates several OASDI tax variables as the sum of corresponding OASI and DI tax variables.

fund Trust fund module calculates fund balance and executes specified asset management strategy.

- **fund_di** DI trust fund module is derived from the trust fund module to perform the DI trust fund calculations.
- **fund_ri** OASI trust fund module is derived from the trust fund module to perform the OASI trust fund calculations.
- **fund_rd** OASDI trust fund module is derived from the trust fund module to perform the OASDI trust fund calculations.

acct Individual account module advances the account balances according to simulated contributions, asset management, and asset returns; converts retirement balances to annuities when specified; and simulates account asset management and withdrawals during retirement years.

ssa The ssa module calculates long-term actuarial program financial measures for OASI, for DI, and for OASDI at the end of each simulation scenario.

olc The olc module contains the high-level logic of the macro model's OLC mode of operation, which requires the optional GEMINI add-on to SSASIM.

esp The esp module contains the high-level logic of the micro model's ESP mode of operation, which requires the optional PENSIM add-on to SSASIM.

rcs The rcs module contains the high-level logic of the micro model's RCS mode of operation, which requires the optional GEMINI add-on to SSASIM.

cohort Cohort module supervises the calculation of cohort lifetime policy performance measures for men, women, and couples using simulation results and the specified cohort policy.

- **policy** Cohort policy module is used by cohort module to specify policy relevant to birth cohort being analyzed.
- **ind** Cohort individual module is used by cohort module to calculate cohort individual policy performance measures.
  - **insured** Insured module is used by ind module to calculate individual's OASDI insured status.
  - **oasdi** OASDI program module is used by ind module to calculate individual's OASDI benefits.
  - **oasdiben** OASDI benefit module is used by OASDI program module to calculate individual's initial OASDI benefit.
- **couple** Cohort couple module is used by cohort module to calculate cohort couple policy performance measures.
  - **marriage** Marriage simulation module generates marital events for individuals.
  - **children** Children simulation module generates life events for individuals' children.

history History module knows historical values of various economic, demographic, and policy variables.

1.5.3 Output Results Files

The model's stochastic simulator program writes as many as 30 macro output results files and up to 29 cohort results files. The actual number written by a run depends on which files are requested in the run's OUTPUT table row. Most results files contain cross-section values of simulated variables for selected years. In these files there is one cross-section observation per simulation scenario in the model run. Other results files contain time-series values for various simulated variables. In these files there is one time-series observation per year for the simulation scenario specified in the run's OUTPUT table row.

The output results files written by the simulator for a model run all have the same file name, but have unique (three character) file-name extensions. The file name itself indicates the model run id number which is a positive integer with no more than five digits. This means that the results file name for run 406 is run00406, while run 1002 would generate a results file name of run01002. If a particular results file had an extension of zzz, then the complete name of that results file would be run99999.zzz for the run with id 99999.

The file-name extensions of the output results files are presented below in the order that they appear in the output table:

asm Average assumption results: post-transition period average of each of thirteen major assumption variables for each scenario.

asx Cross-section assumption results: value of each of thirteen major assumption variables in specified year for each scenario.

ast Time-series assumption results: value of each of thirteen major assumption variables for each year of the specified scenario.

pop Population results: values of various demographic variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

pod Population totals for each calendar year for each scenario.

bid Native births and net immigration for each birth cohort for each scenario.

mrd Mortality rate details: values for age/gender-specific mortality rates for each year of each scenario.

lab Labor market results: values of various labor force variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

prd Product market results: values of various price indexes for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

est Establishment results: values of various productivity, output, and earnings variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

esd Establishment time-series detail results: annual values of various growth model variables for the specified scenario.

cap Capital market results: values of various asset returns for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

cad Capital market time-series detail results: annual values of various equity return variables for the specified scenario.

bnd DI benefit results: values of various benefit variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

bnr OASI benefit results: values of various benefit variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

ben DI and OASI beneficiary counts for each calendar year for each scenario.

pfp PIA formula parameters: bendpoints and formula fractions for each eligibility event for each calendar year for each scenario.

txe OASDI taxable earnings results: values of various taxable payroll variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

txd DI tax results: values of various tax variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

txr OASI tax results: values of various tax variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

tfd DI trust fund results: values of various trust fund variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

dfd DI trust fund annual results: values of various trust fund variables for each simulation year after the initial year in one specified scenario.

tfr OASI trust fund results: values of various trust fund variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

dfr OASI trust fund annual results: values of various trust fund variables for each simulation year after the initial year in one specified scenario.

tfc OASDI trust fund results: values of various trust fund variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

dfc OASDI trust fund annual results: values of various trust fund variables for each simulation year after the initial year in one specified scenario.

act Individual account results: values of various account variables for the initial year and the years 2010, 2020, . . . , 2080 for each scenario.

ssa SSA summary results: values of various long-run summary variables for each scenario.

sti Annual values of the thirteen key assumption variables and the three asset returns for each year of each scenario.

sto Annual values for income rate, cost rate, and trust-fund ratio, for combined OASDI program and for OASI and DI programs separately, for each year of each scenario.

pay Annual values for payable-benefits percent for OASI and DI programs for each year of each scenario.

coh Cohort results: various policy performance measures for a selected number of men, women, and couples who are members of specified birth cohort for each scenario. The cohort results are written to seven sets of files: (1) results for whole cohort to a .coh file; (2) individual and couple cross-references to a .cxh file; (3) female individual results to a set of .cFf files, where F=1,. . . ,9; (4) male individual results to a set of .cMm files, where M=1,. . . ,9; and (5) couple results to a set of .cFM files, where F=1,. . . ,9 and M=1,. . . ,9.

1.5.4 Map of Tables, Modules, and Files

The input database tables, simulator logic modules, and output results files have been listed and briefly described above. This section concludes with a map that associates with each module all the input tables it reads and all the output files it writes. In this map the simulator modules are listed in the same order as they are listed in Section 1.5.2. The map is useful for determining which module's source code to read in order to learn more about the contents of an input table or an output file. The source code is available on request to SSASIM users. The map is contained in Table 6.

Table 6: ***Structural Map of Stochastic Simulator.*** *The input database tables read by and output results files written by each program logic module of the stochastic simulator program.*

Input Database Table Name	Stochastic Simulator Module Name	Output Results File Extension
QUEUE	run	
RUN	run	sti sto pay
OUTPUT	output	
ASSUMP TRND_MV TRND_CC DEV_CC DEV_P	assump	asm asx ast
POP POP_DEL POP_GA0 POPDYN FR_A IMM_GA MDR_GA SP_PROB SP_AGED	pop	pop ast pod mrd bid
LABMRKT LABDYN LFPR_GA UNR_GA EARN_GA	labmrkt	lab
PRDMRKT	prdmrkt	prd
ESTAB	estab	est esd
CAPMRKT	capmrkt	cap cad
	ben	

continued on next page

continued from previous page

Input Database Table Name	Stochastic Simulator Module Name	Output Results File Extension
BEN_DI NBEN_DI ABEN_DI BDYN_DI RBEN_DI INCR_DI RMRT_DI RECR_DI RBSF_DI	ben_di	bnd ben pfp
BEN_RI NBEN_RI ABEN_RI BDYN_RI RBEN_RI INCR_RI DECR_RI RBSF_RI	ben_ri	bnr ben pfp
	ben_rd	
TAXEARN COVERED MTE	tax	txe
TAX_DI TAXR_DI	tax_di	txd
TAX_RI TAXR_RI	tax_ri	txr
	tax_rd	
	fund	
FUND_DI IPAA_DI STB0_DI	fund_di	tfd dfd
FUND_RI IPAA_RI STB0_RI	fund_ri	tfr dfr

continued on next page

continued from previous page

Input Database Table Name	Stochastic Simulator Module Name	Output Results File Extension
FUND_RD IPAA_RD STB0_RD	fund_rd	tfc dfc
ACCT ACCTBAL ACCTAPY ACCTDYN ACCTAA	acct	act
	ssa	ssa
OLC	olc	
ESP ESP_JOB	esp	
RCS RCS_JOB	rcs	
COHORT	cohort	coh cxh
POLICY AIME PIA DIVTIME PIADYNR PIADYNS PIADYND MINPIA MFB MFBDYN ETEST OAIBEN SIBEN WWBEN WPBEN DWBEN DIBEN	policy	

continued on next page

continued from previous page

Input Database Table Name	Stochastic Simulator Module Name	Output Results File Extension
INDACCT		
ACCTBEH	cohort	
INDS IND EARNPCT	ind	cFf cMm
	insured	
	oasdi	
	oasdiben	
	couple marriage children	cFM
HISTORY HISTMR HISTMDR	history	

2 Installing the Model

In earlier versions of the SSASIM Guide, this section explained how to install the model. Installation instructions for SSASIM and the other PSG models can now be found in the "Installation of PSG Models" section of *Getting Started with the PSG Models* ⟨http://www.polsim.com/psghelp.pdf⟩.

3 Running the Model

This section points the reader to documentation on how to specify model runs, execute them, and analyze their output. It all concludes with a discussion of more advanced output analysis methods.

3.1 General Toolkit Capabilities

For a general introduction to running the model, read *Getting Started with the RSF Toolkit* ⟨http://www.polsim.com/rsfhelp.pdf⟩.

For a general introduction to analyzing macro model trust-fund solvency output, start the TFS Toolkit from the RSF Toolkit Analyze menu and read *Getting Started with the TFS Toolkit* available by pressing the Toolkit Help button.

For a general introduction to analyzing representative sample output produced by the micro model, start the EDA Toolkit from the RSF Toolkit Analyze menu and read *Getting Started with the EDA Toolkit* available by pressing the Toolkit Help button.

For a general introduction to analyzing exemplary individual output produced by the micro model, start the ECI Toolkit from the RSF Toolkit Analyze menu and read *Getting Started with the ECI Toolkit* available by pressing the Toolkit Help button.

The rest of this section describes other more advanced output analysis methods.

3.2 Automated Output Analysis

This part of the section provides an introduction to the output analyzer programs that are distributed with the model. In addition to a brief description of each program, examples are provided of their use in scripts to automate post-simulation analysis of output results. As a preliminary, the format of the output results files is described.

3.2.1 Output Results File Format

All SSASIM output is written to results files in the directory where the input database is located. The first three letters of all the output results files are run followed by the run number expressed in five digits. So, for example, the results file name for run number ten would be run00010 followed by a dot and a three-letter extension that identifies the kind of output results included in the file.

A complete list of the different kinds of output results files is included in the OUTPUT table. The Analyze Output Results dialog box has a documentation button that describes each variable in an output file.

Most of the output files are organized in the following manner. The first column lists the scenario for which the output measures are being reported. Remaining columns provide values for the output measures themselves. The first row lists the headings for each of the columns. A few output files contain time-series results for a single specified scenario. In these files the first column contains the simulation year number.

In both file format cases, the column entries in output result files are tab-delimited, which facilitates the importing of these files into either a spreadsheet or statistical package. The advantages of such importing are substantial. The package will usually be familiar to the policy analyst using the model and have a comprehensive range of numerical computation and data visualization capabilities. And perhaps most important is that all such packages can be programmed to perform a complex sequence of tasks. This programmability creates the opportunity to automate the post-simulation analysis of several results files generated by a model run. Such automated processing could produce a series of standard graphs or tables that summarize certain aspects of the run's result. Comparing such summarized results across model runs provides a powerful way to grasp the program finance, contribution return, and benefit adequacy implications of alternative policy regimes.

The output results files are also well suited for processing by a short sequence of one-line AWK programs (Aho, Kernighan, and Weinberger 1988) that can perform, for example, complex analysis of the pattern of benefit receipt across the lifetime scenarios experienced by cohort individuals and couples. A high quality AWK interpreter for Windows is available from the Free Software Foundation.

Yet another option for post-simulation analysis of the output results files is to use one or more of the "res" output analyzer programs distributed with SSASIM. Interactive use of these analyzer programs is provided via the Analyze Output Results item on the Output menu. A brief description of each program is provided before discussing how to use them in a script that automates post-simulation analysis.

3.2.2 Output Analyzer Program: resdist

The resdist program analyzes results by calculating summary statistics for any output variable. The summary statistics include the mean and standard deviation of the output variable values, as well as the 5th, 10th, 15th, . . . , 85th, 90th, and 95th percentile values, if there are at least one hundred observations on the output variable (that is, if the run that generated the results called for at least one hundred scenarios). There is an option to provide more detail, which produces all 99 percentile values.

3.2.3 Output Analyzer Program: resmask

The resmask program analyzes results by producing the same statistics for a specified output variable as does the resdist program, except that, instead of using all the scenario observations, only scenarios in which the value of a masking variable exceeds a specified level are used in the statistical calculations. As with the resdist program, the percentile values of the output variable are shown only if there are at least one hundred scenarios in which the masking variable exceeds its specified level. Note that the output variable and the masking variable may be the same variable or different variables.

3.2.4 Output Analyzer Program: resprob

The resprob program analyzes results by calculating the fraction of the scenarios in which a specified output variable is less than or equal to a specified threshold value. This capability permits estimation of the probability of an output variable being no greater than a certain value.

3.2.5 Output Analyzer Program: rescount

The rescount program analyzes results by counting the number of scenarios in which the value of the output variable falls into each one of many histogram bin ranges. The width of the bin ranges used in the histogram can be specified.

3.2.6 Output Analyzer Program: rescorr

The rescorr program analyzes results by calculating the correlation coefficient between two output variables. This capability permits estimation of the correlation across Monte Carlo scenarios between two different output results variables.

3.2.7 Output Analyzer Program: resgain

The resgain program analyzes results by computing the complete scenario gain distribution and using this gain distribution to calculate a certainty-equivalent (or risk-adjusted) gain value using cumulative prospect theory (Tversky and Kahneman 1992). There is also an option to use expected utility theory (with a specified power utility function risk aversion parameter) in the calculation of the certainty-equivalent of a gain distribution that is always positive (von Neumann and Morgenstern 1944).

Both the complete gain distribution and the certainty-equivalent gain value are written out by the resgain program unless one of the optional final command-line parameters (either one-line or gain-only) is used. If one-line is used, the gain distribution is not written as output, while the results of the certainty-equivalent calculations are written on one line. If gain-only is used, only the gain distribution is written out. In this gain-only case, if the gain distribution results are saved to a file, then other output analyzer programs can be used to analyze the resulting gain distribution or the resulting gain distribution can be imported into a spreadsheet or statistical package for further analysis.

3.2.8 Output Analyzer Program: resce

The resce program calculates a certainty-equivalent value for a specified variable in a result file, using either an expected utility theory power utility function or a cumulative prospect theory value function. In both cases, actual probabilities are used to compute an expected utility or expected value.

3.2.9 Output Analyzer Program: reseqr

The reseqr program calculates the arithmetic mean, volatility, and geometric mean of the corporate equity rate of return found in the runNNNNN.sti output results file. This utility is useful when setting the TRND_MV table parameter values for the eqr assumption variable.

3.2.10 Using the Output Analyzer Programs in a Script

All these SSASIM output analyzer programs can be started from the Windows command prompt by using the command-line syntax documented in the SSASIM Output Analyzer item on the Help menu. This same syntax documentation can be seen from the command line by invoking an output analyzer program with the single command-line option help. So, for example, to see the documentation on the resgain program, simply enter `resgain help` at the command line.

The main advantage of being able to run the output analyzer programs from the command line is that they can be used in scripts (or batch files) that automate complicated post-simulation analysis.

A simple example illustrates how the supplied output analyzer programs may be used in a Windows batch file with the batch file's output being imported into a spreadsheet for visualization. Suppose one wanted to produce a graph of the cumulative distribution function of the equity return in 2050. The strategy illustrated here is to use a batch file to execute the resprob program numerous times and to redirect the one-line output of each execution to a text file that can be imported into a spreadsheet.

First, use a text editor (for example, the Windows Notepad application) to create a file called res_cdf.bat that contains the following lines:

```
@echo off
resprob %1 %2 -30 one-line
resprob %1 %2 -20 one-line
resprob %1 %2 -10 one-line
resprob %1 %2   0 one-line
resprob %1 %2 +10 one-line
resprob %1 %2 +20 one-line
resprob %1 %2 +30 one-line
resprob %1 %2 +40 one-line
resprob %1 %2 +50 one-line
resprob %1 %2 +60 one-line
resprob %1 %2 +70 one-line
```

This batch file is executed at the Windows command prompt using the same first two command-line parameters called for by the resprob program. To produce the cumulative distribution function for the 2050 equity rate of return (which is in the 44th column of the .cap output results file) in run 2, the batch file would be executed by entering

```
res_cdf run00002.cap 44
```

at the Windows command prompt. The eleven sets of results, each formatted on one line, will be shown on the screen.

These results can be imported into a spreadsheet by redirecting the res_cdf results to a text file and importing that text file into the spreadsheet. The easiest way to do this is to execute the following two commands

```
res_cdf run00002.cap 44 > eqr2050.xls
eqr2050.xls
```

at the Windows command prompt (assuming the Excel spreadsheet is installed on the computer). Following these two commands, the eleven lines of res_cdf output will appear as eleven rows in the spreadsheet ready for whatever charting or graphing treatment is desired.

A second example shows how to calculate histogram data for the gain in the present value of lifetime OASI benefits (PVB), where the gain is measured as the difference between the PVB in one policy regime minus the PVB experienced by the same cohort couple or individual under another (reference) policy regime. Such a graph represents the distribution of gains in a comprehensive measure of benefit adequacy. The first step in automating this post-simulation analysis is to create a flexible script for this purpose.

Create a text file called pvbghist.bat that contains the following lines:

```
@echo off
resgain run%1.%3 4 run%2.%3 4 gain-only > x
rescount x 4 %4
```

Now, if a histogram (with bin ranges of 50 thousand dollars) of the PVB gain distribution for the cohort couple consisting of cohort female 3 and cohort male 7 is desired under runs 456 and 556, with run 456 serving as the reference policy regime, then execute the script by entering the following line

```
pvbghist 00556 00456 c37 50
```

at the Windows command prompt.

Such PVB gain histograms can be produced for several cohort couples or individuals by repeatedly calling the pvbghist.bat script from another higher-level script. So, for example, if there are three couple of interest (c77, c88, and c99), then create a text file called pvbghall.bat that contains the following lines:

```
@echo off
call pvbghist %1 %2 c77 50 > pvbghist.c77
call pvbghist %1 %2 c88 50 > pvbghist.c88
call pvbghist %1 %2 c99 50 > pvbghist.c99
```

Now execute the higher-level script by entering

```
pvbghall 00556 00456
```

at the Windows command prompt. Notice the use of the call command when executing a script (rather than a program) in another, higher-level script.

3.3 Automated Policy Search

The final part of this section provides an introduction to the automated policy search capability and the three command-line utility programs that are used to implement this capability. In addition to a brief description of each of these three utility programs, a complete example of automated policy search is provided.

Before describing these three utility programs and the example, a brief introduction is provided to the automated policy search capability. The concept of "policy search" is finding an OASDI policy parameter value that produces some desired value for an output results statistic. For example, suppose one wanted to find the OASI payroll tax rate for years 2010 and beyond that produces a 75-year OASDI actuarial balance of zero. One could repeat executing a run looking for the value of that policy parameter that produces a zero actuarial balance. This new capability simply automates that process. This capability, therefore, does not add any new features to SSASIM; it simply automates what a model user could always do by hand. The automation is accomplished by the user writing a short AWK program, as described below in the example. The AWK program can specify an policy parameter in the SSASIM input database as the focus of the search and can specify any statistic in the output results files as the target of the policy search.

3.3.1 Result Searching Utility Program: resroot

The resroot utility program initiates the automated policy search process as shown in the example beginning on page 55. The resroot program uses a policy parameter scale variable called x and a custom-written AWK program that returns the value of the target output statistic as a function of the value of x — in mathematical terms f(x). The resroot program find the value of x that implies f(x)=0, which in mathematics is called finding the root of the f(x) function; hence the resroot name of the utility program.

The resroot utility program uses root bracketing and bisection to find the value of x that implies f(x) equals zero. For more information on the resroot utility program, access the utility's help feature by entering the following line

```
resroot help
```

at the Windows command prompt.

3.3.2 Database Table Utility Program: dbtable

The dbtable utility program can delete, insert, update, or select rows in SSASIM input database tables that are located in the current working directory. This table manipulation capability is needed by the automated policy search capability.

The dbtable utility program is meant to be used in the AWK program specified as the first resroot command line parameter. For more information about how to use the dbtable utility program, access the utility's help feature by entering the following line

```
dbtable help
```

at the Windows command prompt.

3.3.3 Result Printing Utility Program: resprint

The resprint utility program can print the value of any statistic, for any scenario, that appears in any SSASIM output results file. This output results printing capability has the option of using a specified offset value that adds to the value of the statistic in the output results file. This optional offset can be used when the target output value is different than zero.

The resprint utility program is meant to be used in the AWK program specified as the first resroot command line parameter. For more information about how to use the resprint utility program, access the utility's help feature by entering the following line

```
resprint help
```

at the Windows command prompt.

3.3.4 Using the Automated Policy Search Capability

The AWK program that is shown in Table 7 on page 56 illustrates the use of the automated policy search capability. It illustrates how to automate the policy search problem described at the beginning of this discussion on page 53.

If that AWK program is entered into a file named sample.awk using the Windows Notepad application, then this particular automated policy search process can be initiated by entering the following line

```
resroot sample.awk 10.8 20.0
```

at the Windows command prompt.

When developing other AWK programs for use with the automated policy search capability be sure to read the help available for each of the three associated utility programs: resroot, dbtable, and resprint.

Table 7: **Contents of SAMPLE.AWK file that illustrates use of the automated policy search capability.**

```
# SAMPLE.AWK program illustrates resroot basics

BEGIN {

# suppress echo of Windows script commands
  print "@echo off"; # must be first print statement

# update OASI payroll tax rate in 2010 and beyond using x
# (note: cal_year=2010 must be the last row with id=79034)
  print "dbtable taxr_ri update 79034 cal_year 2010 pay_t_rate " x;

# insert run in QUEUE table
# (note: run 1001 uses TAXR_RI.id=79034 rows)
  print "dbtable queue insert 1001";

# start simulator in silent mode
  print "call ssasim silent"; #OR: print "ssasimx silent";

# print negative OASDI actuarial balance, which is search target
  print "resprint run01001.ssa 1 24"; #24 = neg OASDI act bal

} # end of BEGIN action block and end of AWK program

# For more help on the AWK language:
# enter 'gawkhelp' (without quotes) for AWK manual
```

References

Aaron, Henry J.; Bosworth, Barry P.; and Burtless, Gary. *Can America Afford to Grow Old? Paying for Social Security*. Washington, DC: The Brookings Institution, 1989.

Advisory Council on Social Security. *Report of the 1994–1996 Advisory Council on Social Security, Volume I*. Washington, DC: U.S. Government Printing Office, 1997.

Aho, Alfred V.; Kernighan, Brian W.; and Weinberger, Peter J. *The AWK Programming Language*. Reading, MA: Addison-Wesley, 1988.

Barro, Robert J. and Sala-i-Martin, Xavier. *Economic Growth*. New York, NY: McGraw-Hill, 1995.

Bosworth, Barry; Burtless, Gary; and Steuerle, Eugene. "Lifetime Benefits and Earnings: Evidence for Representative Social Security Participants" and "Lifetime Earnings Patterns, the Distribution of Future Social Security Benefits, and the Impact of Pension Reform." Papers presented at the Social Security Administration's conference on New Developments in Retirement Research. Washington, DC: May 20–21, 1999.

Campbell, John Y.; Lo, Andrew W.; and MacKinlay, A. Craig. *The Econometrics of Financial Markets*. Princeton, NJ: Princeton University Press, 1997.

Cairns, Andrew J.G. "Uncertainty in the Modeling Process." Department of Actuarial Mathematics and Statistics working paper, Heriot-Watt University, Edinburgh, 1994.

Citro, Constance F. and Hanushek, Eric A., eds. *Improving Information for Social Policy Decisions: The Uses of Microsimulation Modeling, Volume I: Review and Recommendations*. Washington, DC: National Academy Press, 1991.

Citro, Constance F. and Hanushek, Eric A., eds. *Assessing Policies for Retirement Income: Needs for Data, Research, and Models*. Washington, DC: National Academy Press, 1997.

Cochrane, John H. “Where Is the Market Going? Uncertain Facts and Novel Theories.” *NBER Working Paper Number 6207*, Cambridge, MA: National Bureau of Economic Research, October 1997.

Coplien, James O. *Advanced C++ Programming Styles and Idioms*. Reading, MA: Addison-Wesley, 1992.

Cotton, Paul and Sadowsky, George. “Future Computing Environments for Microsimulation Modeling,” in Constance F. Citro and Eric A. Hanushek, eds., *Improving Information for Social Policy Decisions: The Uses of Microsimulation Modeling*, *Volume II: Technical Papers*. Washington, DC: National Academy Press, 1991.

Date, C. J. *Database: A Primer*. Reading, MA: Addison-Wesley, 1983.

Diamond, Peter A. “What Stock Market Returns to Expect for the Future?” *Issue Brief Number 2*, Boston, MA: Center for Retirement Research, September 1999.

Fair, Ray C. *Testing Macroeconomic Models*. Cambridge, MA: Harvard University Press, 1994.

Foster, Richard S. “A Stochastic Evaluation of the Short-Range Economic Assumptions in the 1994 OASDI Trustees Report.” Social Security Administration Actuarial Study No. 109, Baltimore, MD: SSA Office of the Actuary, August 1994.

Geweke, John. “Bayesian Comparison of Econometric Models.” Colin Clark Lecture, Australian Meetings of the Econometric Society, July 1994.

Hammersley, J.M. and Handscomb, D.C. *Monte Carlo Methods*. London, UK: Chapman and Hall, 1964.

Holmer, Martin R. “Demographic Results from SSASIM, a Long-Run Stochastic Simulation Model of Social Security,” in *Report of the 1994-1995 Advisory Council on Social Security, Volume II*, Washington, DC: U.S. Government Printing Office, 1996.

Holmer, Martin R. “Stochastic Simulation of Economic Growth Effects of Social Security Reform,” in Olivia S. Mitchell, ed., *Prospects for Social Security Reform*, Philadelphia: University of Pennsylvania Press for the Pension Research Council, 1999.

Holmer, Martin R. *GEMINI Guide*. Washington, DC: Policy Simulation Group, various dates. The most recent version is available at ⟨http://www.polsim.com/doc/guide2.pdf⟩

Holmer, Martin; Janney, Asa; Cohen, Bob. *PENSIM Overview*. Washington, DC: Policy Simulation Group, various dates. The most recent version is available at ⟨http://www.polsim.com/doc/overview.pdf⟩

Lee, Ronald D. and Carter, Lawrence R. "Modeling and Forecasting U.S. Mortality." *Journal of the American Statistical Association*, 87:659–671, September 1992.

Steuerle, C. Eugene and Bakija, Jon M. *Retooling Social Security for the 21st Century: Right and Wrong Approaches to Reform*. Washington, DC: Urban Institute Press, 1994.

Trustees of the Federal OASDI Trust Funds. *Trustees' Report on the Federal Old-Age and Survivors Insurance and Disability Insurance Trust Funds*. Washington, DC: U.S. Government Printing Office, March 1999.

Tversky, Amos and Kahneman, Daniel. "Advances in Prospect Theory: Cumulative Representation of Uncertainty." *Journal of Risk and Uncertainty*, 5:297–323, 1992.

von Neumann, John and Morgenstern, Oscar. *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press, 1944.