

# MPRA

Munich Personal RePEc Archive

## The “Randomizer” Program: Procedure and Operating Methods

Omid M. Rouhani

UC Davis, California Energy Commission

January 2013

Online at <https://mpa.ub.uni-muenchen.de/53540/>

MPRA Paper No. 53540, posted 10 February 2014 15:20 UTC

# The “Randomizer” Program: Procedure and Operating Methods

Omid M. Rouhani<sup>1</sup>

<sup>1</sup> Corresponding Author, School of Civil and Environmental Engineering, Hollister Hall, Cornell University, Ithaca, NY, 14853, USA.; Email: [om67@cornell.edu](mailto:om67@cornell.edu), Tel: +1 530 204 8576

## ABSTRACT

This report provides a detailed explanation of the “Randomizer” program. The “Randomizer” program calculates the benefits of Electric Program Investment Charge (EPIC) research projects by estimating the effects of the projects on some of the parameters incorporated in the RPS Calculator. The “Randomizer” program is written in the VB environment inside an Excel spreadsheet. The program determines the random effects of EPIC projects on the model’s parameters. Values for the effects on the parameters are drawn repeatedly from one of the few statistical or empirical distributions. Once a set of values are drawn for all the involved parameters, the modified RPS Calculator is run. The main modification is related to the demand responsiveness to electricity prices. Finally, changes in electricity prices, carbon emissions, and renewable portfolios from the base case (without EPIC projects) are reported as the outputs of the program.

## RPS CALCULATOR MODEL

The RPS Calculator (RPS Calculator website, 2012) is a Microsoft Excel spreadsheet-based model developed to select renewable resources needed to meet the RPS target(s), aggregating the renewable costs and performance data.

The RPS calculations start with a statewide calculation of the renewable resources that California utilities must procure between 2008 and 2020 to meet a predetermined RPS target by 2020. The resources needed are then calculated as the total required quantity of renewable energy in 2020 minus the actual renewable generation claimed by California utilities in 2007.

Projects are ranked and using a project scoring system. The project scores are based on a combination of cost, environmental, and commercial considerations. Then, for each zone, resources and projects are organized into bundles according to their scores and transmission constraints. The procedure continues until the specified RPS target is met.

Based on the mentioned procedure, the RPS Calculator estimates California’s annual electricity expenditures and portfolios of renewables from 2009-2020 under each scenario (Rouhani et al., 2013).

The “Randomizer” program uses the RPS Calculator as its basic model. Two major modifications are applied to the RPS Calculator. First, electricity demand is incorporated as fixed forecasts adopted from the CEC (2009) study with different basic assumptions (Results in inconsistency). Our model incorporates short-run and long-run demand functions, that are responsive to prices, and estimates the demand endogenously. Second, all the parameters/ inputs of the RPS Calculator can be randomized, and changes in the parameters can be made using various distributions.

This report explains the way to run the program, the used method for applying the above mentioned changes, the VB code written for this purpose, and the computation time needed for running the program.

### **HOW TO RUN THE “RANDOMIZER” PROGRAM**

The first step to running the program is to “enable Macros”. To do this, click the “Options” button once the “Security Warning, Macros have been disabled” text appears. The text appears every time you open the Excel file. Then, choose “Enable this content”. Now, the VB macros can be viewed or run.

The program can be run using two different methods. The first is to click the “Run Randomizer Model” button inside the "PIER Instructions" sheet. Before running the model, you should add all Epic Projects and parameters affected by the EPIC projects and all project information needed for performing the run.

The second method is to run the program by opening the VB code. To do this, you should choose the “View” tab in the Excel file, click the “Macros” (right side of the window), and select “View Macros.” A new window pops up listing all the VB macros that have been created for the model. The program/macro created for our purpose is called “Randomizer”. You can choose to “Step into” or “Run” the macro. When it is running, the “Randomizer” macro makes use of all the other macros. If you choose to step into the macro, the “Randomizer” code will be opened. Then, you can run the model inside the VB environment (using the Run tab or by clicking ▶). The benefit of choosing the second method is that in addition to running the model, you can see the code and make necessary changes to it.

### **PROCEDURE**

Figure 1 shows the different steps of the “Randomizer” Visual Basic (VB) program. The program is organized in seven different steps:

**Step 1:** Declare/Define main parameters.

**Step 2:** Input the number of draws-to-make (ND) & define the main sheets.

**Step 3:** Clear the results of previous runs & run RPS Calculator once.

**Step 4:** Input the electricity demand-related parameters.

**Step 5:** Find a random draw for each random parameter, iteratively.

**Step 6:** Update demand forecasts iteratively. Go to step 5 until all simulation draws are made.

**Step 7:** Report outputs/results of running the modified RPS for each set of draws.

The VB program is attached in the Appendix using the above colors code and with clarifying comments. The general procedure (Figure 1) is as follows: the program starts by defining major variables (step 1-light blue) and assigning values to some inputs (step 2-yellow), such as the number of simulation draws (ND). Then, the results from the previous runs are removed (step 3-green). Next, the demand functions' parameters and demand elements' shares are read (step 4-purple).

Then, the first iteration ( $ND_r=0$ ) is used to calculate the results of the base case (without EPIC projects). The next iterations ( $0 < ND_r \leq ND$ ) are used to calculate the effects of the random changes EPIC projects make to the RPS calculator parameters (inputs). Each iteration is run using new random draws (the changes in the exogenous parameters of the RPS Calculator). For each iteration, the random changes resulting from all EPIC projects are found, using "GenRandom" function defined in the VB code (step 5-Red). Then, the RPS Calculator is run iteratively to match the electricity demand and electricity prices with each other (step 6-dark green). After finding a closely-related demand and price outcomes, the results of the set of simulation draws are reported in the "PIER Outcome Draws" sheet (step 7-gray), and the procedure continues until all draws are made ( $ND_r=ND$ ).

As shown in Figure 1, the program consists of two major loops: the first loop for demand and price synchronization, and the second loop for iteratively running the model using different sets of simulation draws. The first loop is designed to find demand values which match electricity prices. The original RPS Calculator operates with fixed annual demand forecasts (CEC, 2009) as exogenous inputs and estimates the electricity prices. Therefore, we iteratively run the RPS calculator using the updated demand levels/inputs (based on the selected demand functions). This procedure can be repeated until the demand values and prices (average costs) converge to corresponding values ( $Demand(\text{iteration } N) = Demand(\text{iteration } N-1)$ ). However, because of limited computation time, we use only 5 iterations as the stopping criterion.

The second loop is intended to iteratively calculate the models' results using different sets of simulation draws. Each simulation draw consists of a set of draws for all the parameters/types of benefits that the EPIC projects affect. These draws are made based on the defined distributions for each type of benefit that can be gained from the project. These random draws then determine the changes in the RPS Calculator parameters or a collection of projects. Once the set of changes in the parameters are determined for each simulation draw, we can run the modified RPS model. Then, the results are saved. The procedure moves to the next iteration, and continues until the iterations' limit is reached ( $ND_r=ND$ ). The procedure ends after reporting all the simulation runs' results.

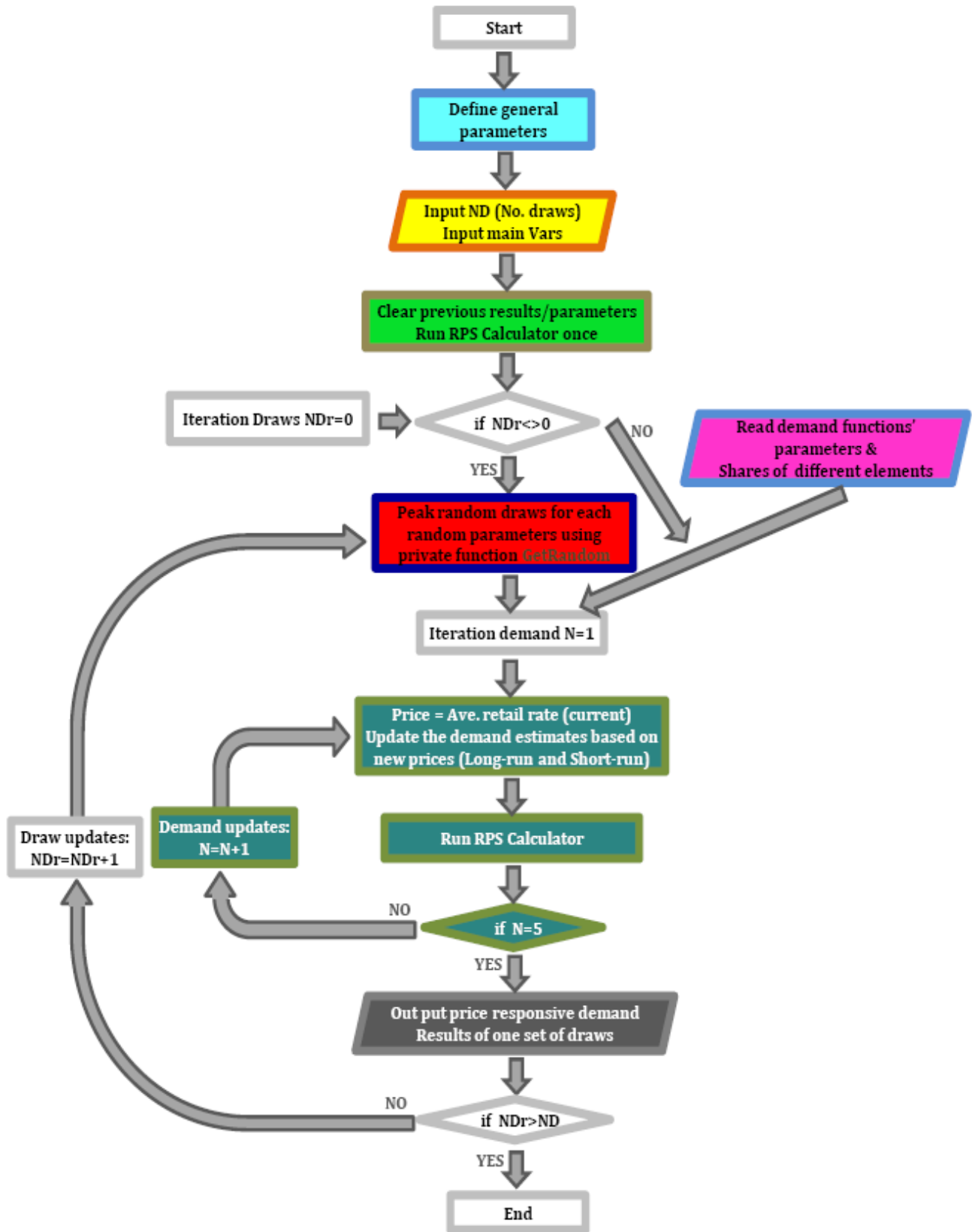


Figure 1 Overall flowchart of the EPIC model

More detailed steps of the program are provided in Figures 2 to 5. Figure 2 shows the detailed procedure for performing the steps 1 through 3. The first step (light blue) defines the main variables used for programming. These variables consist of the number of draws (“Draws”) and number of projects (“Lines”) variables, parameters of each distribution, location of the distribution parameters, integer variables (“i”, “j”, “k”) mainly used for forming loops, and the number of empirical distributions (“empiricalnumbers”). The distribution parameters include the type of distribution (“Distribution”), “Upperbound”, “Lowerbound”, and “CentralEstimate”. Also, the location-columns of these parameters are defined as variables (“DistributionCol”, “UpperboundCol”, “LowerboundCol”, and “CentralEstimateCol”).

The second step (yellow) assigns the number of draws’ value to the corresponding variable (“Draws”-ND). Also, the main sheets ("PIER Projects to Model", "PIER Parameter Draws", "PIER Outcome Draws") are named so that referencing these sheets will be easier. Finally, the columns’ numbers containing distribution parameters are stored in the corresponding variables.

The third step (green) deletes the previously saved values inside the “PIER Outcome Draws” sheet and the current parameter (Column H) of the “PIER Parameter Draws” sheet. The deletion restores the original conditions/parameters so that the results of the base case (without EPIC projects) can be calculated. Then, the RPS Calculator is run once, and the main results are stored.

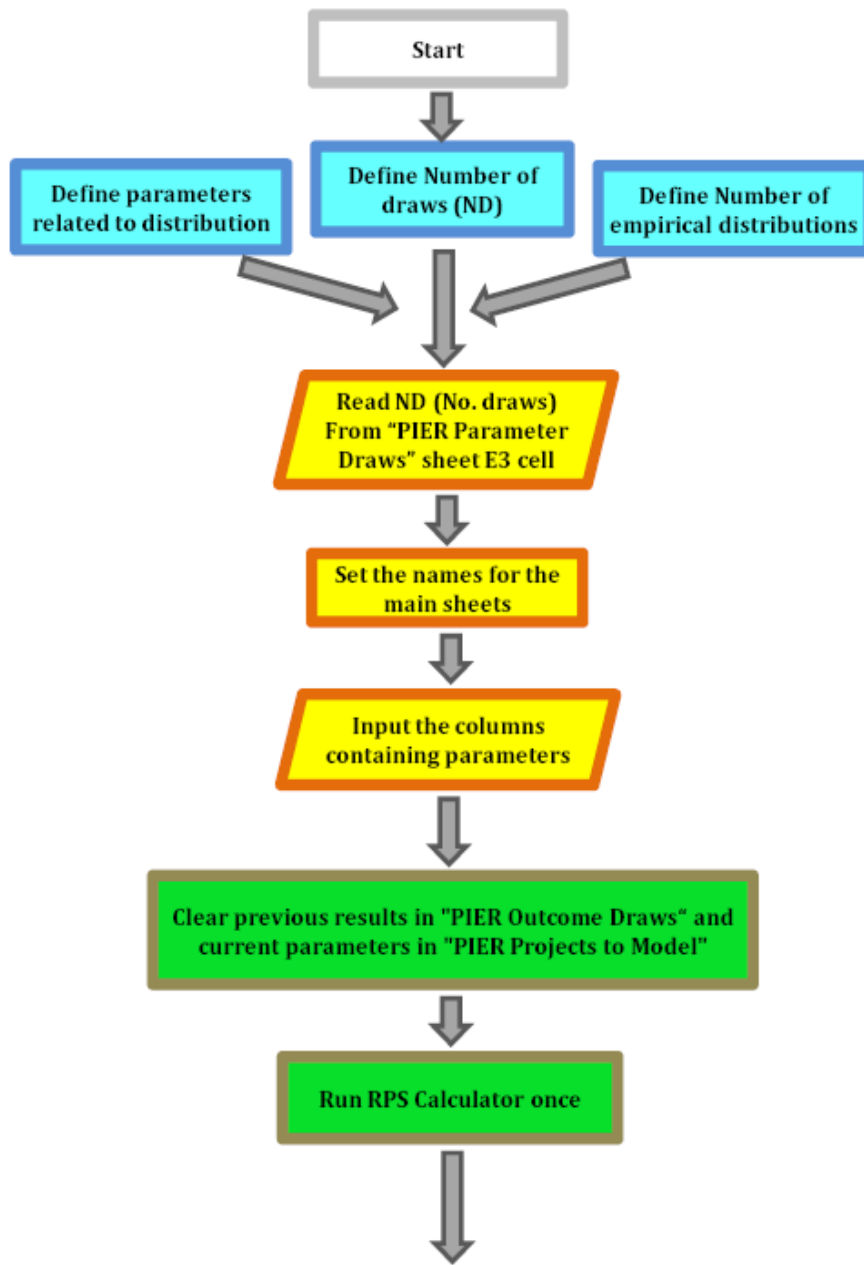


Figure 2 Steps 1 to 3 of the procedure

Figure 3 details the procedures used for steps 4 and 5. Step 4 (purple) begins with defining the electricity demand functions' parameters (elasticity, A-coefficient), and the demand related variables (electricity prices in different years - Array "Electricity-prices" - and total electricity demand- Array "NetEnergy"-). Then, values from the Excel sheets are assigned to these variables. Demand functions' parameters are read from the "PIER Parameter Draws" sheet-cells H15 to H18.

Step 4 continues with defining and assigning the shares of different elements of electricity demand in 2012 and 2020, such as Peak share of total demand, Peak CSIDG share, Peak energy efficiency reduction share, Peak electric vehicle share, Coincident Peak share, as well as total energy efficiency reduction share, total electric vehicle share of total , and end-use share of total. Next, the shares of these variables for the 2013 to 2019 period will be calculated using interpolation. Also, values are assigned to the shares of demand for providers. Finally, the demand response variables are stored.

Step 5 (red) is designed to find random draws for each of the randomized parameters/types of benefits affected by PIER projects. Note that we can also randomize parameters which are not affected by EPIC projects, but we are not sure of the exact value of those parameters (stochastic variables, e.g. fuel prices can be considered as distributions instead of one fixed number).

For the first iteration (NDR=0), parameters are fixed at their original (non-random) values so that the base case (without EPIC effects) results can be calculated. For the next iterations, random draws for the randomized parameters (types of benefits) should be found. To this end, the information about all parameters' distribution are read from the "PIER Projects to Model" sheet.

Then, random draws are iteratively made using the "GenRandom" function. These draws will be stored in Item Draw cells (Column M) of the "PIER Projects to Model" sheet. The shifts in total demand and peak demand are stored separately. The demand shifts are not used directly for changing the inputs/parameters. The shifts should be applied to the demand functions, and the final demand values will be used to update demand inputs/forecasts in the Excel sheets.

Once all the benefits are drawn randomly, the set of new parameters can be calculated and used for a new RPS Calculator run. The run first updates all the parameters by adding the effect of EPIC projects to the original value and then replaces the existing values with the new ones.



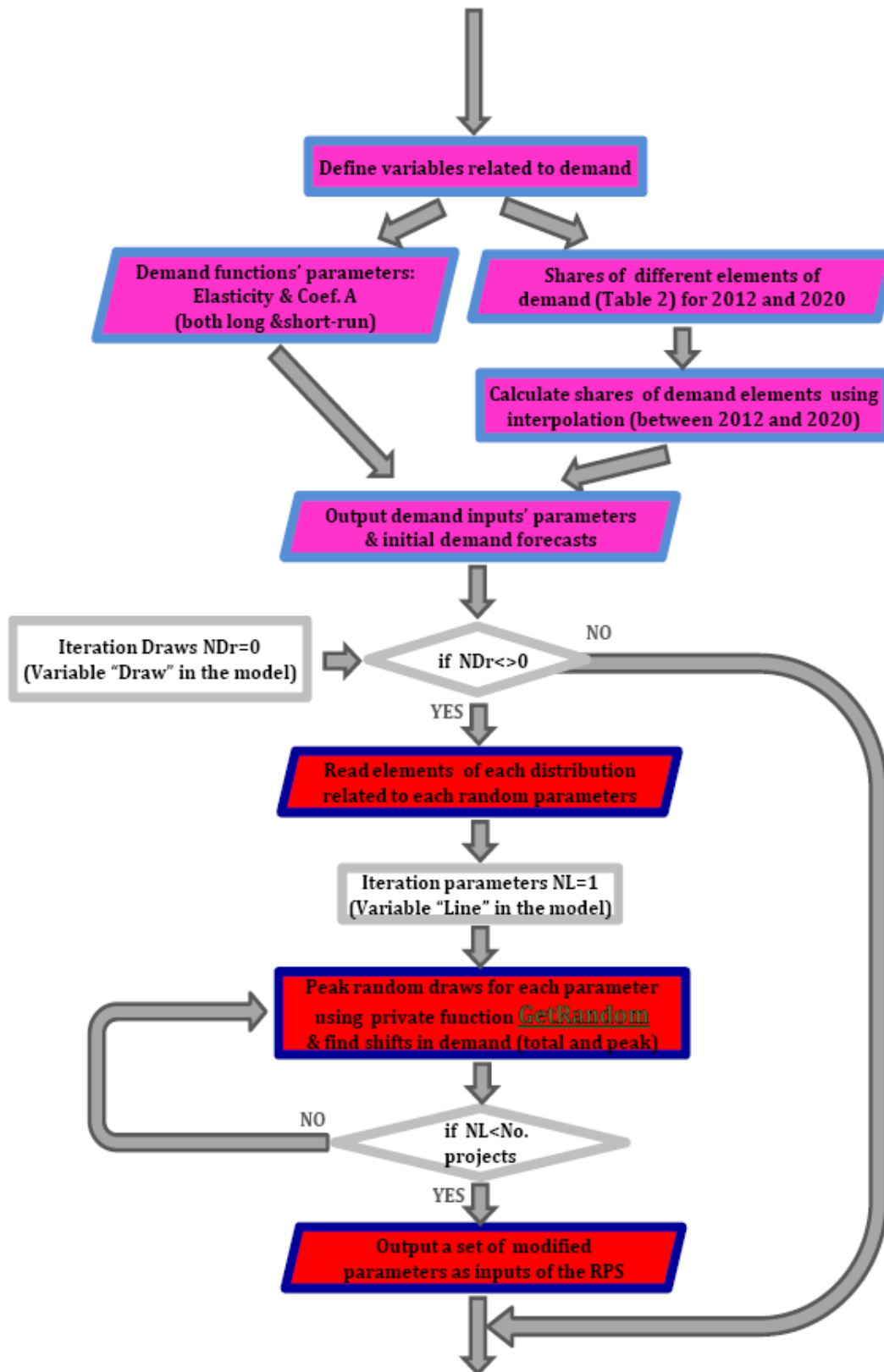


Figure 3 Steps 4 to 5 of the procedure

Figure 4 illustrates the steps of the “GenRandom” function. VB programs can make use of newly defined functions. These functions are used in order to perform a repetitive action without repeating the accompanied code unnecessarily. The GenRandom function is defined to pick a random draw for each parameter (type of benefit). To do so, the mean (“Estimate”), type (“Distribution”), potential value (“PotentialType”), lower bound (“LowerBound”), upper bound (“UpperBound”) of distributions should be added as inputs of the function. Simulation draw number (“DrawNo”) is also an input of the function. The “DrawNo” variable determines the draw (row) of the empirical distributions which should be selected.

Two approaches are employed for creating random draws. One is to select one of the existing distributions (Log-Normal, Normal, and Uniform) and choose a random number from the distribution. The other is to use empirical distributions and directly get the random draw from the empirical distribution list. The list has been added in Column AC and the following columns from the “PIER Projects to Model” sheet. The later approach only needs the simulation draw number to pick the values consecutively from the list. The output of the function is a random draw for each parameter/type of benefit.

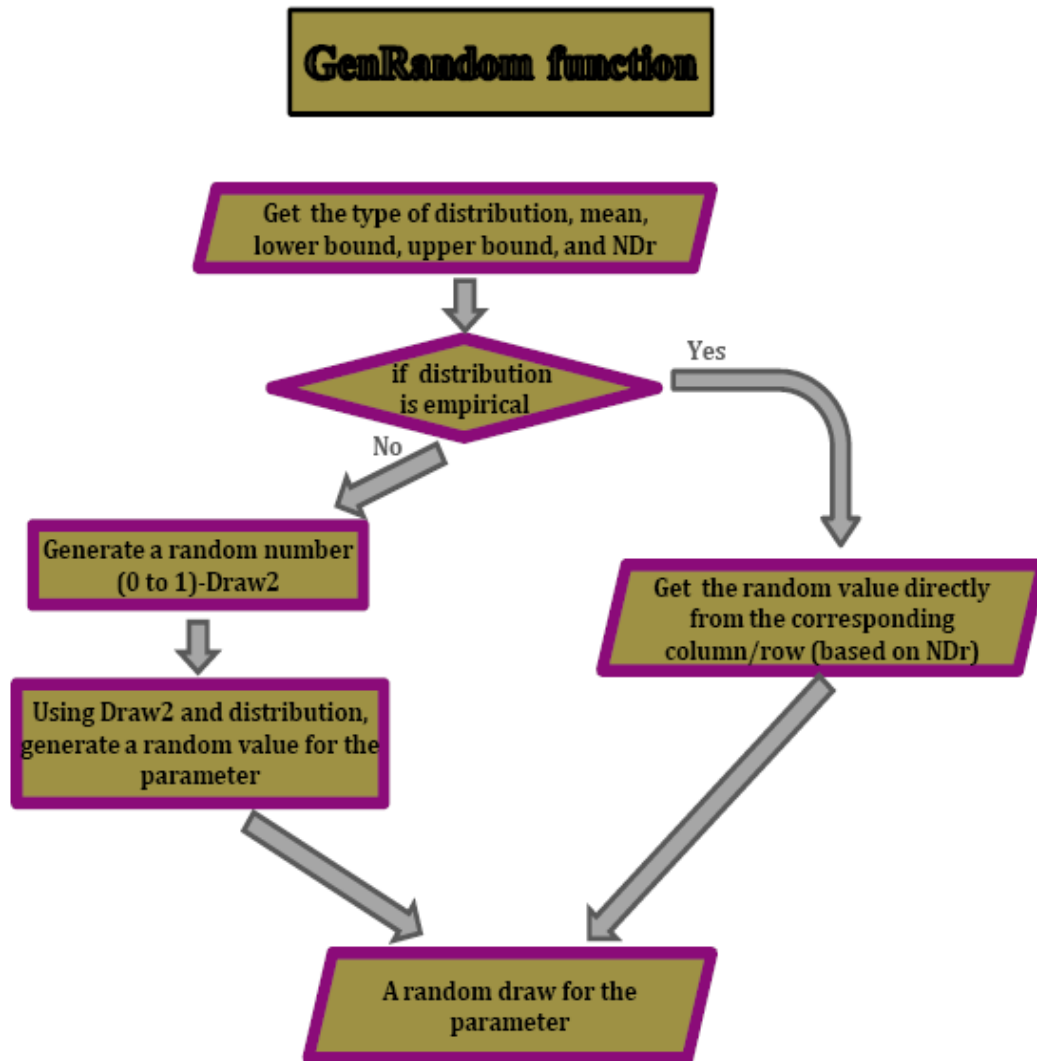


Figure 4 Gen random function flowchart

Figure 5 shows the steps 6 and 7 of the program. Step 6 (dark green) is devised to match electricity prices with electricity demand levels iteratively. For the first iteration, we should run a VB Macro called “ResourceCalc”. This Macro updates the costs of generating electricity for each type of generation based on new technology costs. We only need to run this Macro once whenever the cost parameters related to generating electricity are changed (only when N=1).

The electricity prices are estimated by the average costs of producing electricity. The average costs can be found from the Row 361 of the ("zz - Cost Impacts") sheet after running the RPS Calculator. The total short-run demand in 2012 and the total long-run demand in 2020 are calculated based on the respective short-run and long-run demand functions, and all elements of demand are being updated using the updated total demand and the shares of each element in each year.

For the first iteration, the average costs of producing electricity (2008\$/kWh) for 2012 through 2020 are estimated by running the RPS calculator with its original demand forecasts. Then, the 2012 and 2020 electricity prices are calculated assuming prices equal the average costs (Kavalec et al., 2012). The prices, then, are used to estimate the 2012 through 2020 total demand based on the short-run and long-run functions, and the demand forecasts are replaced by the new calculated demand values. For the second iteration, the RPS calculator is run again but with the updated demand values, which results in new average costs, new prices, and new demand values consequently.

This procedure can be repeated until the demand values and prices (average costs) converge to corresponding values ( $\text{Demand}(\text{iteration } N) = \text{Demand}(\text{iteration } N-1)$ ). However, we choose five iterations of demand updating, which in most cases results in no more than 1% deviation in the demand values of the last two iterations. The outputs of Step 6 are the demand, prices, and costs forecasts between 2012 and 2020 for a set of simulation draws. The procedure should be repeated for all simulation draws until all the simulation draws are made ( $N_{Dr} = N_D$ ).

Step 7 (gray) reports the important results of running the RPS calculator. The results include average retail rates-electricity prices in different years (Row 361 of the "zz - Cost Impacts" sheet), 2020 in state and out of state renewable generation (Cells K4 to M12 of the “z – Results” sheet), annual energy demand, annual total cost or revenue requirement (Row 360 of the "zz - Cost Impacts" sheet), annual LCA change in GHG emissions, 2020 change in labor, and annual non-LCA GHG mitigations. These results are saved in the “PIER Outcome Draw” sheet.

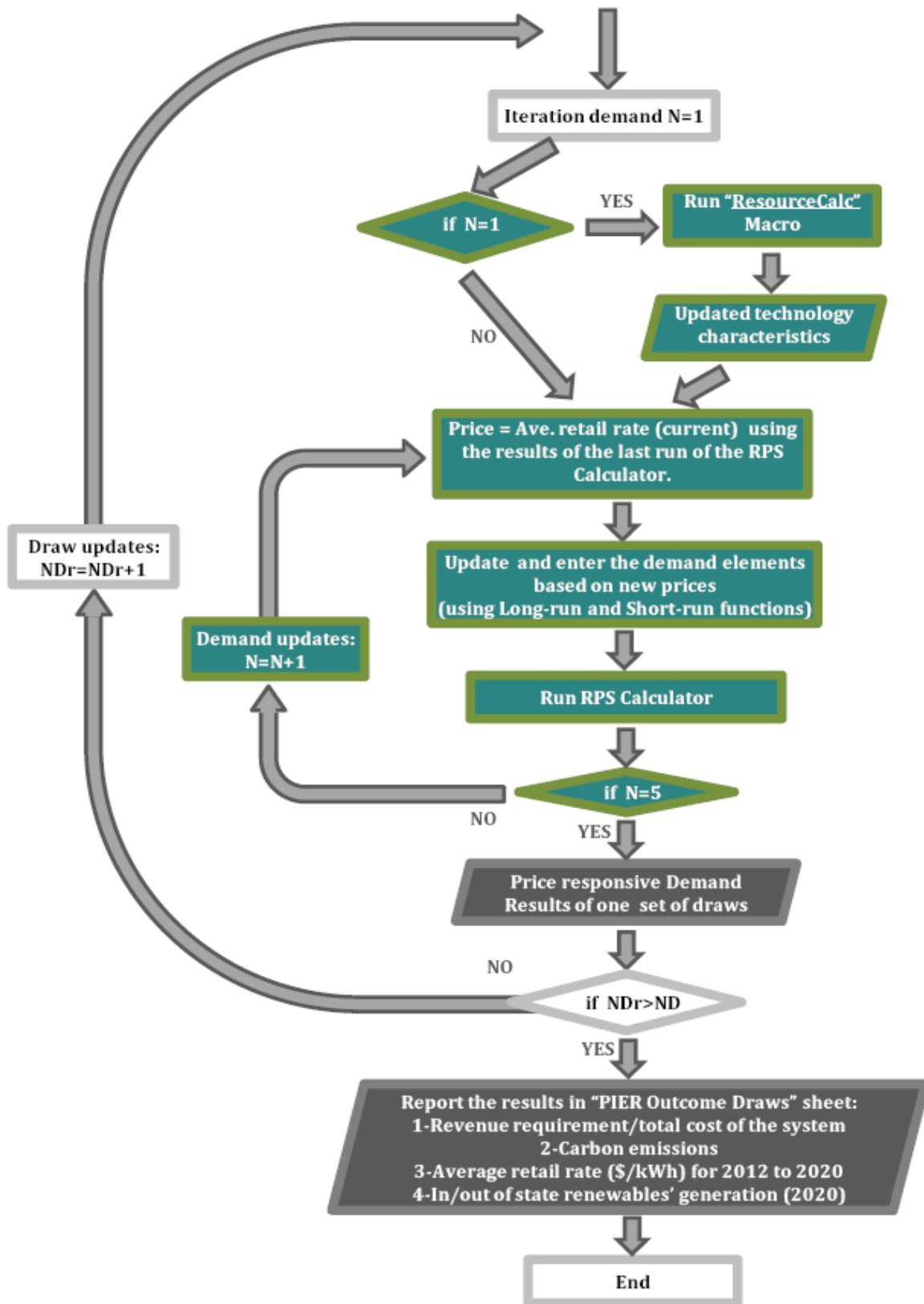


Figure 5 Steps 6 to 7 of the procedure

## **COMPUTATION TIME AND ERROR**

The addition of the responsive demand feature to the RPS Calculator leads to higher computation time and some errors. The computation time of the original RPS calculator is about 9 seconds, so the computation time of the modified model is about 45 seconds (for one simulation draw, iterating 5 times). For some of our runs, which might need 100 to 1000 randomized simulations, the total computation time will be about 75 minutes to 12.5 hours, respectively.

The error in the calculation stems from the iterative process and indicates a difference between demand values in two consecutive iterations (the demand and price relationships in the demand functions and the intrinsic relationship inside the model do not match). Excluding Case 6, the highest difference in demand between two final iterations is 0.7% of total demand, which is a relatively small number. The exceptions are the cases which involve big changes in fuel costs. We might consider adding more iterations to the procedure in some cases, especially for high gas prices.

One last note about the computation time is that when some of the parameters related to the costs of different generation types are changed, a VB program (macro) called “ResourceCalc” must be run to update the levelized cost of the generations (as shown in Figure 1). This macro takes about 1.5 minutes, and significantly increases the computation time. As a result, the total computation time of performing 100 to 1000 randomized simulations increases to 3 hours and 45 minutes to 37.5 hours, respectively.

The respective computation time of the alternative version of the program is over 19 hours. The computation time is based on reporting 500 results (5 values for 100 projects). The time can change using more/less projects or potential values.

## **COPYRIGHT NOTE**

The author certifies that he has the right to deposit the contribution with MPRA.

## REFERENCES

### ***Model reference:***

RPS Calculator website (2012 version). Available online at:

<http://www.cpuc.ca.gov/PUC/energy/Procurement/LTPP/2012+LTPP+Tools+and+Spreadsheets.htm>

### ***Input data References:***

Kavalec, C., Fugate, N., Gorin, T., Alcorn, b., Ciminelli, M., Gautam, A., Sharp, G., Sullivan, K. (2012). California energy demand forecast 2012-2022 volume 1: statewide electricity demand and methods. California energy commission, electricity supply analysis division. Publication Number: *CEC-200-2012-001-SF-VI*, 2012. Available online at: <http://www.energy.ca.gov/2012publications/CEC-200-2012-001/CEC-200-2012-001-SF-V1.pdf>, Accessed August 2012.

California Energy Commission, California Energy Demand 2010-2020 Adopted Forecast. 2009. Available online at:

<http://www.energy.ca.gov/2009publications/CEC-200-2009-012/CEC-200-2009-012-CMF.PDF>

### ***Other references:***

Rouhani, O.M., Kandel, A., Christian, M. (2013). The California Renewable Portfolio Standard's Impacts on the Electricity Sector in an Uncertain Cost Environment. *International Journal of Power and Energy Systems*, 3(3), 130-134.