Replication steps for *Tax Credits for Clean Electricity: The Distributional Impacts of Supply Push Policies in the Power Sector*

Maxwell Brown

March 2024

1 Introduction

The files included in this package are those necessary to re-create the work referenced in the title. Generally, the user should be familiar with GAMS as well as Python in order to re-create the linked model structure. As an overview of the process for running the models, a user needs to (a) run the benchmark cases for the ReEDS-SLiDE linkage, (b) run the initial three tax credit policy scenarios based off their respective benchmark cases, (c) run the two carbon cap scenarios using the TC IBA emissions results and based off their respective benchmark cases, (d) run the demographic disaggregation model from the SLiDE results, then (e) run the InMAP model using the ReEDS emissions results.

ReEDS functions as the flagship model for running the link ReEDS-SLiDE model as well as creating the power sector results. ReEDS is an electricity capacity expansion model that minimizes the cost of power system investment and operations while stepping forward one year at a time in a myopic fashion. The model is open-source and there are several publicly-available resources to learn how to run ReEDS, such as the following YouTube playlist which covers the core model as well as several of its modules:

ReEDS model YouTube playlist

In addition, in the ReEDS directory as well as at its public-facing GitHub page is a lengthy readme which summarizes how to set up the software environment on various operating systems and has a large list frequently asked questions asked by users:

ReEDS model GitHub repo

It is advisable to become comfortable running ReEDS before attempting to run the linked version with SLiDE. While SLiDE only requires GAMS, ReEDS has several pre- and post-processing steps that require a specific software environment to be set up. We have tried to make this process as easy as possible via

issuing a conda environment that will install and utilize the necessary Python packages.

2 Computing requirements

As a general guideline, the recommended system for running a single scenario with the linked model is as follows:

- 8 threads of Intel 12th generation or later (or equivalent AMD) CPU
- 80GB of RAM
- 30GB of storage space

For software requirements, only two programs are needs:

1. GAMS (v38+) - see: GAMS homepage
2. Anaconda (most-recent version) - see: Anaconda homepage

Both of these software packages should be able to called from the terminal via adjusting the PATH environment variable.

The following licenses are required:

- GAMS base module
- GAMS CPLEX [1]
- GAMS PATH
- GAMS MPSGE

With this general setup, one can expect the linked case to solve with convergence in approximately 48 hours. For the work published here, we have developed the modeling framework to run on a Linux-based high performance computing cluster with the SLURM task management system. Although we generally attempt to make cross-platform compatibility possible, it is not guaranteed.

[1] Other linear programming solvers can be used but ReEDS has not been tuned to non-CPLEX solvers; attempts to use open-source solvers have been unsuccessful.

3 Main file descriptions

- **environment.yml** - contains the information of the Anaconda- and pip-based packages needed to re-create the data pre- and post-processing steps. The environment is typically created by running the command: *conda env create -f environment.yml*. Note the most-recent version of Anaconda should generally be reverse compatible with the versions of python and python packages referenced in the environment.yml file. The environment create from that step can be loaded via the command *conda activate reeds*.

- **runbatch.py** - this processes the information necessary to create the various scenarios. Running the script and following thge prompts will copy over the minimum files necessary to begin a call to the model. In addition, the model will create either a .sh or .bat file that will contain the commands for running the various models. Importantly, the options given to write out those instructions are constained in the cases.csv (or cases_X.csv where X=[your chosen file name]) files.

- **cases.csv** - This contains the (many) options for running a ReEDS scenario; in addition, it also determines what switches are passed to SLiDE via the reference to the appropriate slide_options.csv file. Note that each script has options across multiple columns, the first being the switch name, the second being a short description of the switch and its options, the third being the available switch settings, and the fourth being the default value for the switch. The most important switches specific to the ReEDS-SLiDE linkage are as follow:

  - *slide_dir* Directory for clone of SLiDE repo
  - *slide_gams* Directory of GAMS to use for calling SLiDE
  - *slide_iter* Number of iterations between ReEDS and SLiDE (+1)
  - *slide_regionality* Regionality for SLiDE

- **cases_irasuite_h17.csv** - this contains the default set of switches for the reference cases (prefixed by REF_....) and counterfactual policy scenarios. Note this file is used across a range of the various policy representations that are determined either in ReEDS (by adjusting the incentives_suffix or GSw_Annual_Cap settings). The remaining options determine the performance and cost trajectories indicated in the appendices of the main paper.

- **slide_options.csv** - This file holds the options that are passed to SLiDE including policy representations, regionality, revenue recycling mechanisms, as well as several other options.

- **createmodel.gms** - This is the core file which creates the ReEDS model by invoking four other files:

  - **b_inputs.gms** - Processes inputs, creates sets/parameters, and generally develops the logic to define the boundaries of the problem. The results of the *input_processing/*.py* scripts are loaded into the model in this file.
  - **c_supplymodel.gms** - Contains the constraints of the ReEDS model
  - **c_supplyobjective.gms** - Contains the objective function of the ReEDS model
  - **d_solveprep.gms** - Prepares for solving

- **d_solveoneyear.gms** - Loads in information from the planning reserve margin and capacity credit calculating submodule, Augur, processing the financial multiplier parameters, determines the tax credit phaseout trajectory, and invokes *d_solveoneyear_link.gms*.

- **d_solveoneyear_link.gms** - loads in data from both the benchmark scenario as well as SLiDE specific to that iterations, solves the QCP, and creates a gdx file that is sent to SLiDE

- **e_report.gms** - processes results and prepares data for post-processing steps.

- Within the SLiDE repository, the main directories of interest for this linkage are contained in [repo base]/models/SLiDE:
  - *bmk_data* - Contains the data necessary for re-creating the benchmark social accounting matrix
  - *data* - Contains further data for dynamics, households, and calibration
  - *gdx*, *logfile*, *lst* - directories created new for each run that contain exchange and results data, the log file of the current run, and the GAMS .lst files for debugging and data displays (respectively)
  - *mod/mgemodel.gms* - contains the MPSGE version of the model
  - *loop* - contains the files necessary for either standalone or linked (via loop_loadlink.gms) solves. The loop_loadlink.gms solve contains the steps for loading in and processing the ReEDS data into SLiDE.

- The *[Demographic_Dissagregation]* directory contains the post-processing files necessary for leveraging the FusionACS-CE data via the ReEDS results - the two files to perform the demographic disaggregation steps are in the [root]/welfare_gams/(rep_st.sh or run_dem_st.bat) files for linux and windows systems, respectively. In addition, the necessary .g00 files (run results) for performing the disaggregation are contained in the [root]/welfare_gams/g00_store directory.

4 Model setup and running

The process for running the model is to:

1. Install all necessary programs as indicated in the ReEDS tutorials and readme.md file including a version of GAMS that can run MPSGE/PATH (this is referenced in the aforementioned slide_gams option in cases.csv).

2. Run a set of benchmark cases that can serve to parameterize the reference price, quantity, and elasticity values in the ReEDS QCP formulation. This is done through creating the options necessary for a benchmark cases in a separate cases_X.csv file (similar to the REF_... cases in the cases_ira_h17.csv file or the cases_standard_scenarios.csv file).
   This is done in the same fashion as any other ReEDS scenario except the 'timetype' switch needs to be set to 'rs' which then prompts runbatch.py to copy over the necessary ReEDS files and SLiDE directory (via the 'slide_dir' setting). All calls to the system will be written to a [base ReEDS repo]/runs/[case name]/call_[case].sh file which will also include the referenced SLiDE options.

3. Run a set of tax credit-based counterfactuals. Generally, the same options from the benchmark case can be used except then the user needs to change the 'incentive_suffix' file to 'ira' in their new cases.csv file as well as mimic the SLiDE-specific settings demonstrated in the cases_ira_h17.csv file. This includes the flags that indicate both (a) this is not a benchmark case and (b) which benchmark case should the model reference for loading in data.

4. Run a set of carbon cap counterfactuals by enabling the 'GSw_AnnualCap' switch and specifying the trajectory through the 'GSw_AnnualCap_Scen' setting. Note the emissions pathways can either be loaded in via the GDX file or specified in the [repo]/inputs/carbonconstraints/co2_cap.csv file.

5 Results files and calculation sources

The data files for all figures are included in the *outputs.zip* archive. The sources of calculations for the table and figure parameters are featured in table on the next page. The primary Tableau workbook and hyper file are included in the *tableau.zip* archive.

Item	Short description	Location of calculation	Parameter names	*outputs.zip* file	Notes on compilation
Figure 3	Capacity and generation for mid-case scenario	e_report.gms; e_report_dump.py	cap_out and gen_out	capgen.csv	Only mid-case results are presented here; automatically called in .sh/.bat file created via runbatch.py
Figure 4	Power sector emissions characteristics	e_report.gms; e_report_dump.py	Emissions = emit_out; tax credit expenditures = expend_out; carbon credit price = credit_price; carbon cap revenue computed as emit_out * credit_price	emit_out.csv, expend.csv, credit_price.csv, and carbon_cap_revenue.csv (respectively)	Results presented for all scenarios; files automatically created through calls written in the .sh/.bat file created via runbatch.py
Figure 5	NPV of EV by Change in Gini Coefficient	SLiDE/include/loop/loop_store_vint.gms; final processing is done in [slide repo]/models/SLiDE/rep_post/unload_slide.gms	gini; wdecomp	EV_vs_Gini.csv	After the run has finished, unload_slide.gms is called for all years and the resulting gdx files, labeled %casename%_%solveyr%, are merged via gdxmerge utility
Figure 6	Equivalent variation, deaths, and combined impacts by household	EV calculated the same as in Figure 5 but using the household index on wdecomp; reduced deaths calculated via InMAP totalPopD parameter in [inmap]/inmaputil/inmap.go; combined impacts calculated in postprocess_for_tableau.py	SLiDE: wdecomp; InMAP: totalpopd, deaths, emis	npv_ev.csv, reduced_deaths.csv, npv_ev_health.csv	EV notes same as Figure 5, combined impacts calculation done in Tableau
Figure 7	NPV of EV by income class, policy, and income source	Same as Figure 5	wdecomp	hh_inc_decomp.csv	Same notes as Figure 5
Figure 8	Map of NPV of distributional impacts	Same as Figure 5	wdecomp for NPV of EV; AllCause and Xmort parameters from InMAP where X = income class or race	map_hh_polc_all.csv	Same notes as Figure 5; InMAP results are combined in Tableau
Figure 9	Change in NPV of EV by demographic group	[demographics repo]/welfare_gams/loaddem_st.gms	dem_welf_pv; dem_welf_st_pv	CSV files in demographic_outputs contain outputs by demographic category (e.g. age, race, . . .)	Commands for creating the output gdx files which are subsequently merged via gdxmerge utility are included in the rep_st.sh and run_dem_st.bat files
Table 3	NPV of impacts including avoided climate damages	e_report.gms; e_report_dump.py; distributional impacts same as Figure 6; SCC taken from literature	emit_out, wdecomp; deaths	[values presented in text]	
Figure 10	Change in Gini coefficient by region	Same as Figure 5	gini_st	gini_st.csv	Same notes as Figure 5
Figure 11	Change in EV by household and year	Same as Figure 5	wdecomp	ev_hh_year.csv	Same notes as Figure 5
Figure 12	Map of NPV of EV by Income class	Same as Figure 5	wdecomp	map_hh_polc_ev.csv	Same notes as Figure 5
Figure 13	Map of NPV of EV by Race/Ethnicity	Same as Figure 9	dem_welf_pv; dem_welf_st_pv	map_race_polc_ev.csv	Same notes as Figure 9