Core Model Proposal #391: GCAM 7.0 Bugfixes

Product: Global Change Analysis Model (GCAM)

Institution: Joint Global Change Research Institute (JGCRI)

Authors: Ellie Lochner, Maridee Weber, Pralit Patel, Matthew Binsted, and Nathan Wiens

Reviewers: Gokul Iyer and Kanishka Narayan

Date committed: June 3rd, 2024

IR document number: PNNL-36105

Related sector: Economy, land, emissions/climate, and core

Type of development: Code

Purpose: Various bugfixes for GCAM 7.0 prior to the GCAM 7.1 release

Description of Changes

<u>gcamdata Testing Framework</u>

A set of changes was made to get the *gcamdata* tests to pass, including fixing capitalization of constants, removing consecutive mutates, switching ifelse into if_else, and replacing rbind with bind_rows. Full details on which chunks were changed and why is included in the table below.

<u>Spelling</u>

We also ran a spell checker on some of the code and made some spelling fixes. There are more errors, mainly in the R code comments that should be fixed over time but are not included in this proposal. Changes are reflected in the table below, indicated by "Spelling".

<u>Dollar Unit Conversion Consistency</u>

In the C++ code, FunctionUtils::DEFLATOR_1990_PER_DEFLATOR_1975() was updated from 2.212 to 2.219 to ensure consistency with the value used in gcamdata::gdp_deflator().

<u>Query Updates</u>

GCAM-USA queries added for "CO2 emissions by tech (nested subsector) (excluding resource production)" and "CO2 sequestration by tech (nested subsector)" to account for nesting subsector structure within GCAM-USA's electricity sector.

Changes in XML names was causing Primary Energy (the CCS version worked fine) and Internal Gains queries to miss results, which has now been corrected.

<u>R 4.4 and Associated Renv Lockfile Update</u>

A recent discovery of security vulnerabilities in R having to do with loading saved .rda (etc.) files has been identified which affects all R versions. Given the way R installs packages behind the scenes this means all R packages could be vulnerable. The vulnerability has been fixed in R 4.4 (as far as I know it hasn't been back ported to previous releases).

We take this opportunity to move gcamdata and upgrade R to 4.4 and all package to the latest versions in our Renv Lock file. Note, this also deals with the issue that recent upgrades to Apple / Xcode's compiler means we are unable to install some of the package versions referenced in our previous Renv Lock file. Finally, just lock file updates were needed. No further changes to gcamdata were needed to address any package compatibility issues.

Note, some changes are made to the testing framework to switch to using R 4.4 on deception as well.

<u>Minor Changes to Accommodate gcamwrapper / compiling GCAM using Rtools on Windows</u>

We add some optional compiler flags to the Makefile, which a user may choose to set as an environment variable, to facilitate successfully compiling GCAM using the Rtools environment and subsequently linking to that in gcamwrapper in R on Windows. Note, given the way R is ported to Windows this is required unlike Python where we can link the GCAM library compiled via Visual Studio directly.

In addition, we add an interface to the GCAM logger to explicitly set an output stream to send "console" messages to. This is a safer way to ensure GCAM's logging messages properly show up in the R / Python console when using gcamwrapper.

Add Feature to Write to a Remote BaseX DB

Writing to a "remote" XMLDB has been a feature we had long speculated we may want however, given we did not have a concrete use case, had been left un-implemented. Our collaborators at KAPSARC did have a concrete use case: the desire to collect results from workers running scenarios concurrently in the cloud into a single BaseX XMLDB.

To do this we add a new Java class WriteRemoteBaseXDB which derives from the existing WriteLocalBaseXDB. The main class XMLDBDriver will check if the user has configured (in the configuration.xml for instance) an xmldb-location using the following URI syntax. If so, it will instantiate the WriteRemoteBaseXDB class instead of the WriteLocalBaseXDB. The URI will be in the format of basex://<username>:<password>@<host>:<port>/<DB Name> . It will then attempt to connect to that URI and stream the XML there. And example configuration could be:

```
<Configuration>
    <Files>
        <Value
name="xmlInputFileName">../input/gcamdata/xml/modeltime.xml</Value>
        <Value name="BatchFileName">batch_ag.xml</Value>
        <Value name="policy-target-
file">../input/policy/forcing_target_4p5.xml</Value>
        <Value
name="GHGInputFileName">../input/magicc/inputs/input_gases.emk</Value>
        <Value write-output="1" append-scenario-name="0" name="xmldb-
location">basex://test:test@localhost:1984/database_basexdb</Value>
```

Of course, a user would be required to be already running a BaseX server to connect to. This is a feature provided entirely by BaseX. Quite similar to connecting to remote XMLDBs in rgcam / gcam_reader we could use the following script to setup and run the server:

```
#!/bin/sh

# A Java classpath that minimally includes BaseX.jar, ModelInterface.jar,
# and BaseX's supporting libs (required to run the HTTP server)
CLASSPATH='/Users/pralitp/models/libs/jars/*:/Users/pralitp/models/gcam-
core/output/modelinterface/ModelInterface.jar:/Users/pralitp/models/libs/base
x/lib/*'

if [ "$1" = "stop" ] ; then
```

```
    # The user just wants to stop an already running server
    java -cp $CLASSPATH org.basex.BaseXHTTP stop
    exit 0
elif [ $# -ne "1" ] ; then
    # we were expecting a single argument which is the path to a folder
    # which does / will contain the BaseX DBs
    echo "Usage:"
    echo "$0 <path to databases>"
    echo "$0 stop"
    exit 1
fi

DBPATH=$1
echo "DB Path: $DBPATH"

# Ensure BaseX users have been set up since remote access will require a
# username and password.  To run Model Interface queries requires READ
access.
# To add data both WRITE and CREATE will be needed.
if [ ! -e "${DBPATH}/users.xml" ] ; then
    echo "No users.xml found in $DBPATH"
    echo "Enter a user name to create one now (or CTRL-C to copy/create a
users.xml manually):"
    read username
    java -cp $CLASSPATH -Dorg.basex.DBPATH=$DBPATH org.basex.BaseX -c"CREATE
USER $username;GRANT WRITE TO $username;GRANT CREATE TO $username"
fi

# Run the server, note only the DBPATH is overriden here, all other settings
are
# to replicate standard GCAM settings, we could experiment with these
# which may create trade off between DB write time / disk space / query time
# Any other settings will pull from the config file in ~/.basex
java -cp $CLASSPATH \
    -Dorg.basex.DBPATH=$DBPATH \
    -Dorg.basex.ATTRINDEX=false \
    -Dorg.basex.FTINDEX=false \
    -Dorg.basex.UPDINDEX=false \
    -Dorg.basex.CHOP=true \
    -Dorg.basex.ADDCACHE=true \
    -Dorg.basex.INTPARSE=true \
    org.basex.BaseXHTTP -d
```

Change CachedMarket to lookup the MarketContainer

Change CachedMarket to lookup the MarketContainer (across periods) instead of Market (specific to a period). This is safer as, especially during reporting, it could have been cached to the wrong period, and no warning or error would have been given (unless running in debug mode). Note: this issue does not come up at present but would for reporting certain details to the XMLDB which has come up in project work.

We also will use CachedMarket as value instead of pointer. Ultimately this is a very minor change, but it will slightly reduce memory and maybe offset some of the pointer chasing added by now having to offset the Market look up by model period.

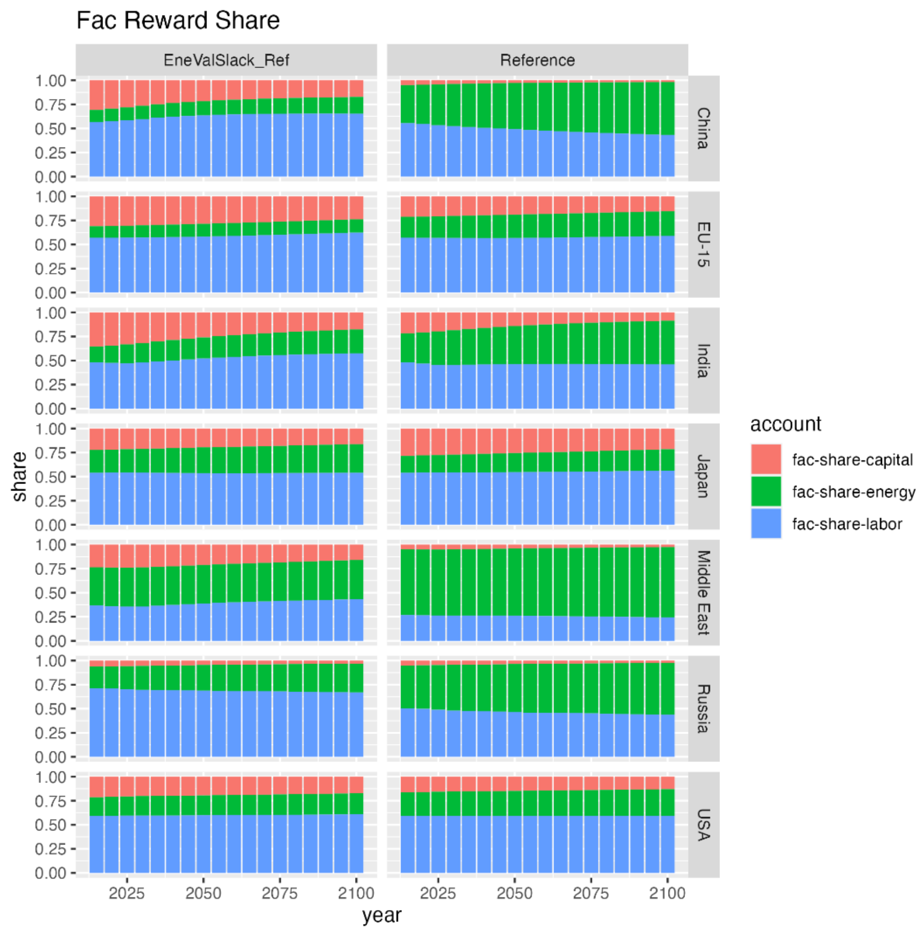
Tweak Negative Price Adjustments in Food Demand Function

Move the negative price adjustment down to a later section of code in the food demand as its current location creates an inaccurate trial share value to be logged into the market. This does not seem to significantly affect any solution performance but could have been related to the generation of messages about "Food budget exceeds total income".

Tweak Factor Share Calibration of Macro CES

To calibrate the nested CES equations of the Materials sector in the Macro model we need to provide it value (dollar) "returns to" Labor, Capital, and Energy. The value for Labor was used directly from our historical National Accounting dataset: Penn World Table. We had originally decided to use GCAM's estimate for Energy value (quantity times GCAM's historical final energy prices) directly and Capital would serve as the remaining "slack" (ensure all values sum to the estimation of gross materials output from Penn World Table / our Social Accounting Matrix). However, given GCAM's lack of regional tuning, particularly final energy prices, and the fact that Penn World Table does provide a value we can use to estimate the Capital value: In this Core Model Proposal we to switch to use the Energy value as the "slack" instead of Capital. Another way to think about this is we are introducing a scaler to match a top down (Penn World Table) and bottom up (GCAM's estimation) of Energy final energy service value instead of relying on the bottom up completely.

This helps generate more realistic value share in many regions, such as China. For other regions, such as USA, it doesn't make much of a difference. Given the Macro model is still run in "fixed" mode it does not affect GCAM results. However, if we were to run in "open" mode we would see some regions such as China to be less sensitive to increased energy prices, while other such as USA are about the same.



Fix bug in ProductionStateFactory When Non-Constant Timesteps

Fix a bug in ProductionStateFactory which was assuming constant timesteps at the vintage's initial year. Now that we have vintaging active (AgStorage) in the historical period (where we switch from 15-year steps to 5), we observe the bug.

Now that the bug is fixed, we can properly calculate (and do error checking of) lifetimes of the AgStorageTechnology and ensure it exactly lasts two model periods. This fixes the original ad hoc approach to setting AgStorageTechnology lifetimes which would have broken if we choose a different model time config.

Fix "Stale" Value Bug in AgStorageTechnology

We had used an IO coefficient in calcCosts which is calculated in calcDemand which occurs after. Granted it was just to back out a value to reset it to 1. So, in fact the end result of the bug is just slightly different round off error on subsequent calls to World.calc. It is simple enough to just set the IO coefficient to 1 for the purposes of the calcCosts function to fix the bug and avoid all round off error.

Fix Units Reported for some National Accounts in the XMLDB

The units for Population and Labor Force were reported as "million people"; it should be "thousand". In addition, per capita GDPs were reported as "million 1990$"; it should be "thousand 1990$ per capita". These are now fixed.

Address Negative Profit rates to Calibrate Ghost Biomass Shares

We solve for a price of biomass in the calibration years. It is one of the few fuels / crops we do not have a calPrice for. This price has fallen in a string of recent GCAM Core Model Proposals including: bugfix on Forest residues (369), explicit modeling of pulp and paper (370), Ag param updates (which shifted the residue biomass curve to the left – 393), and Residential Consumers which re-classified some biomass as Traditional Biomass in the historical data (362).

It has become an issue because it means the calibration profit rate used to calculate the "ghost shares" for biomass have gone negative for a handful of basins, so we cannot calculate a share weight for them.

As background, "ghost shares" are related to the procedure we use to calibrate parameters (share-weights) to introduce new crop (or technology) where they were not present in the historical data, namely biomass. At a high level the procedure compares the profit rate (price - cost) of the new crop to the profit rate of the dominant crop in the final calibration period. And we supply as assumption the "ghost share" which is, given the relative economics, the share we think the new crop would receive within that land nest, i.e., share of crop land. The model then calibrates the share-weight to be used in future model periods.

Long-term it would be ideal if we can have regional estimations for the calPrice of biomass. But given the heterogenous nature of the fuel it is not straightforward.

Instead in this Core Model Proposal we address the conceptual issue: negative profit rates during calibration should not be un-expected (if purpose grown bioenergy were profitable then presumably we would see it grown in the historical period). From this perspective we modify our ghost share calibration to include the precondition "if costs come down X%" then this is what we expect the ghost shares to be. Mechanically, we reduce the nonLandVariableCost by N% in the final calibration year only (when we calculate our "ghost share weights"). And to really realize that share the price would have to go up (or costs go down) proportionately. This is a subtle difference, but it gets us past the mechanical constraint of not being able to calibrate around negative profit rates.

Here we propose a 10% nonLandVariableCost reduction for biomass in the final calibration period only. Which is sufficient to avoid the negative profit rates we have been observing. It makes surprisingly little difference on the whole. We can see the share weights move in the right direction, but again small and that translates into only minor changes in broader results: biomass prices, land allocation, and production.

An example of calibrated "ghost share weights" in the current Core Model:

```
<LandNode name="biomassTree_MexCstNW">
        <land-allocation>0</land-allocation>
```

```
                                <profit-rate>1.44173e+06</profit-rate>
                                <share>0</share>
                                <share-weight>0</share-weight>
                                <ghost-unnormalized-share>-0</ghost-
unnormalized-share>
                                <is-ghost-share-relative>1</is-ghost-share-
relative>
                                <share-weight year="2025">0.00713796</share-
weight>
                                <ghost-unnormalized-share
year="2025">0.01</ghost-unnormalized-share>
                                <share-weight year="2030">0.0160604</share-
weight>
                                <ghost-unnormalized-share
year="2030">0.0225</ghost-unnormalized-share>
                                <share-weight year="2035">0.0249828</share-
weight>
                                <ghost-unnormalized-share
year="2035">0.035</ghost-unnormalized-share>
                                <share-weight year="2040">0.0345001</share-
weight>
```


And following the 10% cost reduction for calibration purposes. We see an increased calibration profit rate, and subsequently lower share weights. But the effect is not large.


```
                            <LandNode name="biomassTree_MexCstNW">
                                <land-allocation>0</land-allocation>
                                <profit-rate>2.01835e+06</profit-rate>
                                <share>0</share>
                                <share-weight>0</share-weight>
                                <ghost-unnormalized-share>-0</ghost-
unnormalized-share>
                                <is-ghost-share-relative>1</is-ghost-share-
relative>
                                <share-weight year="2025">0.00709546</share-
weight>
                                <ghost-unnormalized-share
year="2025">0.01</ghost-unnormalized-share>
                                <share-weight year="2030">0.0159648</share-
weight>
                                <ghost-unnormalized-share
year="2030">0.0225</ghost-unnormalized-share>
                                <share-weight year="2035">0.0248341</share-
weight>
                                <ghost-unnormalized-share
year="2035">0.035</ghost-unnormalized-share>
                                <share-weight year="2040">0.0342947</share-
weight>
```


Note in CMP 382 we had to introduce a "stop gap" which detected the negative profit rates, reset the share-weight to zero, and generated the warning "Negative or invalid profit rate for calibration of <region>, <crop>". Prior to this fix, NaN values were generated causing entire land nests to produce zero output. If economics change in future work, we could once again see the "Negative or invalid profit rate" warnings, however, do not ever expect to see the NaN land allocation issue again.

Description of Code Changes

This table details the code chunks and input files that were modified or added. TF stands for testing framework and indicates a change made to get gcamdata tests to pass.

Chunk	Reason
constants.R	TF: Capitalizing constants correctly
data.R	Spelling
dstrace.R	Spelling
module-helpers.R	Spelling
xml.R	Spelling
zaglu_L100.regional_ag_an_for_prices.R	TF: Capitalization, consecutive mutates
zaglu_L108.ag_Feed_R_C_Y.R	TF: Capitalization
zaglu_L110.For_FAO_R_Y.R	TF: Capitalization
zaglu_L123.LC_R_MgdPastFor_Yh_GLU.R	TF: Capitalization
zaglu_L162.ag_prodchange_R_C_Y_GLU_irr.R	TF: if_else
zaglu_L181.ag_R_C_Y_GLU_irr_mgmt.R	Spelling
zaglu_L202.an_input.R	TF: Capitalization, if_else
zaglu_L203.ag_an_demand_input.R	TF: Capitalization
zaglu_L221.land_input_1.R	Spelling
zaglu_L240.ag_trade.R	TF: Capitalization
zaglu_L2012.ag_For_Past_bio_input_irr_mgmt.R	TF: Capitalization, consecutive mutates (hack to fix because they really aren't consecutive)
zaglu_L2042.resbio_input_irr_mgmt.R	Spelling, TF: Consecutive mutates, if_else, Capitalization
zaglu_L2052.ag_prodchange_cost_irr_mgmt.R	TF: Capitalization
zaglu_L2242.land_input_4_irr_mgmt.R	Spelling
zemissions_L102.nonco2_ceds_R_S_Y.R	Spelling
zemissions_L142.pfc_R_S_T_Y.R	Spelling

zemissions_L2112.ag_nonco2_IRR_MGMT.R	Spelling
zenergy_L115.roofPV.R	Spelling
zenergy_L118.hydro.R	Spelling
zenergy_L132.industry.R	Spelling
zenergy_L225.hydrogen.R	Spelling
zenergy_L1092.iron_steel_GrossTrade.R	TF: bind_rows, if_else
zenergy_L1231.elec_tech.R	Spelling
zenergy_L1323.iron_steel.R	TF: if_else
zenergy_L2323.iron_steel.R	TF: bind_rows, if_else
zgcamusa_L101.EIA_SEDS.R	Spelling
zgcamusa_L154.Transport.R	Spelling
zgcamusa_L223.electricity.R	Spelling
zgcamusa_L245.water_demand_municipal.R	Spelling
zgcamusa_L1235.elec_load_segments.R	Spelling
zgcamusa_L1321.cement.R	Spelling
zgcamusa_L2232.electricity_FERC.R	Spelling
zsocio_L101.Population.R	Spelling
zwater_L232.water_demand_manufacturing.R	Spelling
driverdrake_vignette.Rmd	Spelling

Emissions Related Changes

Outlier Emission Factor Replacement Function (GCAM-USA)

To consistently replace any outlier emission factors (EF) outside of a threshold with a median EF, the **replace_outlier_EFs** function was created in **module-helpers.R**. This function calculates a national (GCAM-USA) or global (GCAM) median EF based on a user-specified grouping, and replaces any NA EFs, Inf EFs or EFs outside the threshold with the median. The threshold is two standard deviations above and three below the median. This function was applied to replace outlier EFs for air pollutants in the GCAM-USA transportation, electricity generation, buildings, and industrial energy use sectors.

Outlier Emission Factor Replacement Function (GCAM)

The **replace_outlier_EF** function was also applied to emission factors calculated by dividing emissions derived from CEDS with energy consumption estimated within gcamdata. In this instance, it was applied to all supplysectors in the CEDS data except for out_resources. The out_resources supplysector was not included since previous work implemented an alternate strategy to address outliers here, as additional processing needed to be done to exclude model regions that had very low resource production and did not yield reliable emission factors.

Maine N2O Emissions from Coal Electricity Generation (GCAM-USA)

Maine was previously assigned all GHG emissions from the 'coal_base_conv pul' technology in 2015 due to a poorly executed left_join. The 'coal_base_conv pul' technology exists in 2015 so that it is there for future periods, but is not used in 2015. The table containing fuel input for electricity generation does not have vintaged technologies, while the table containing electricity generation output does have vintaged technologies. When these tables were joined, all of the entries that get matched for 'coal_base_conv pul' have 0 calibrated output, except for in Maine. In all other states, the output is distributed between vintaged technologies (e.g., 'coal_base_conv pul 1991-1995'). We addressed this by distributing emissions across all of the 'coal_base_conv pul' technologies in each year based on the calibrated coal consumption.

GCAM-USA Industrial Process Emissions (GCAM-USA)

Process emissions from industry were driven by population growth, which lead to an unreasonable increase in emissions in the future. We now reduce future emissions from industrial processes inverse to population growth.

Iron and Steel (GCAM)

Previously, emissions in the iron and steel sector were input driven which resulted in unreasonably high emissions for certain regions like Africa_Southern, which has zero iron and steel production in 2015. We changed emissions in the iron and steel sector to be output driven, and to use emission factors rather than input emissions. By switching to output driven emission factors, we can ensure that regions with zero iron and steel production do not have emissions from this sector.

Description of Code Changes

This table details the code chunks and input files that were modified or added.

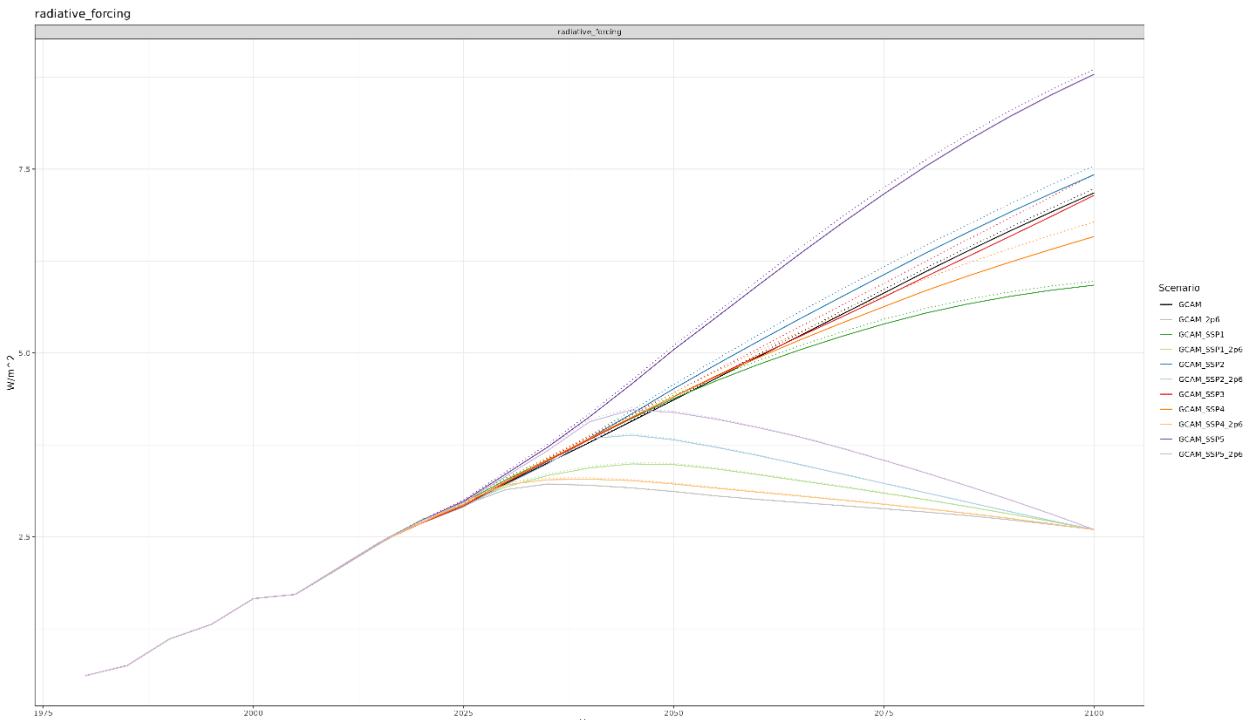
Chunk	Description of Changes
module-helpers.R	Added function "replace_outlier_EFs".

zgcamusa_L271.nonghg_trn.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L272.nonghg_elc.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L274.nonghg_bld.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L275.nonghg_indenergy.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L276.nonghg_othertrn.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L2722.nonghg_elc_linear_control.R	Code changes to implement the "replace_outlier_EFs" function.
zgcamusa_L231.proc_sector.R	Code changes to add a control that keeps emissions from growing in future years due to population increase.
zgcamusa_L2236.elecS_ghg_emissions.R	Code changes to address a bug where GHG emissions were not accounting for vintaged technologies. Previously, this resulted in large N2O emissions in Maine for electricity generation from coal.
zemissions_L112.ceds_ghg_en_R_S_T_Y.R	Code changes to implement the "replace_outlier_EFs" function.
zemissions_L241.en_newtech_nonco2.R	Code changes to implement the "replace_outlier_EFs" function.

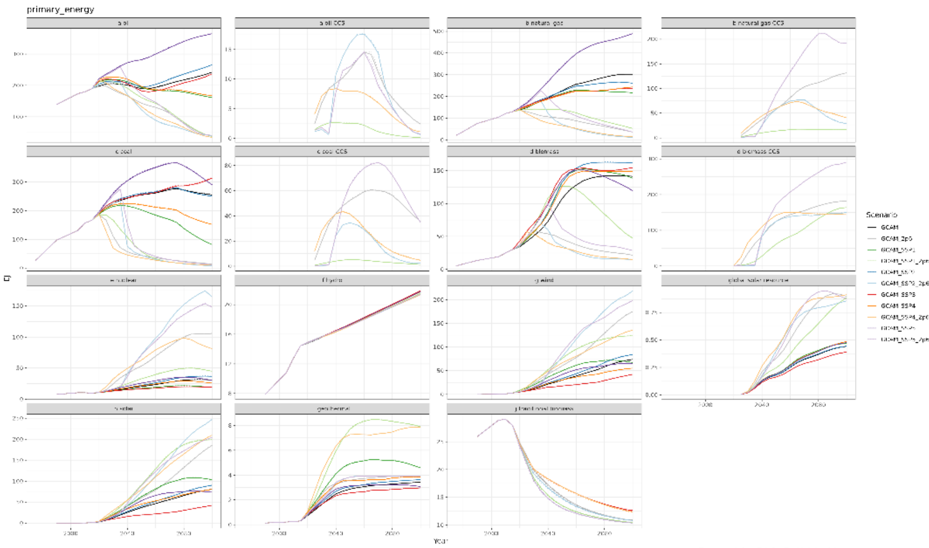
zemissions_L201.en_nonco2.R	Code changes to make the iron and steel sector have output driven emission factors, and implement the "replace_outlier_EFs" function.
zemissions_xml_all_energy_emissions.R	Added iron and steel output emission factor table to XML.

Validation

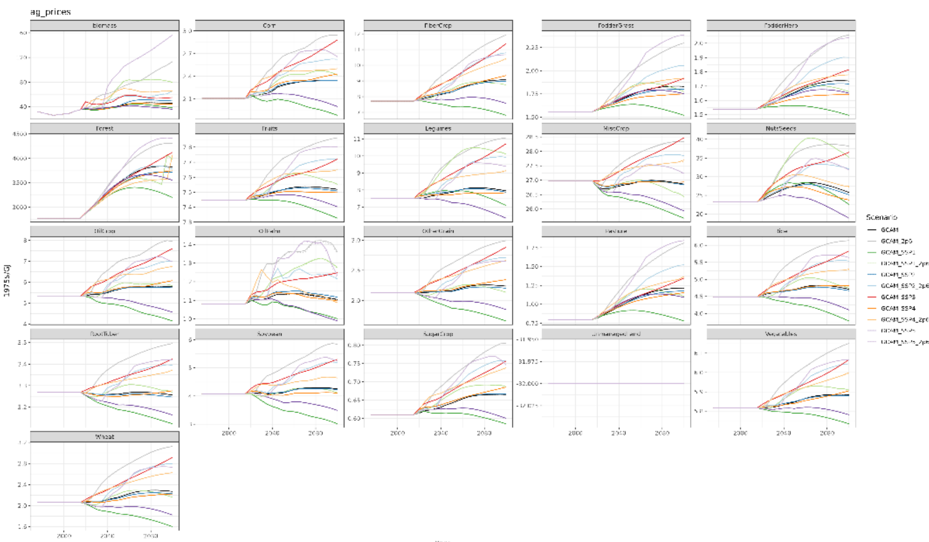
Results mostly do not change with the exception of the non-CO2 emissions factor outlier fixes. Which can have a big impact on some species, however, does not impact overall results too much.



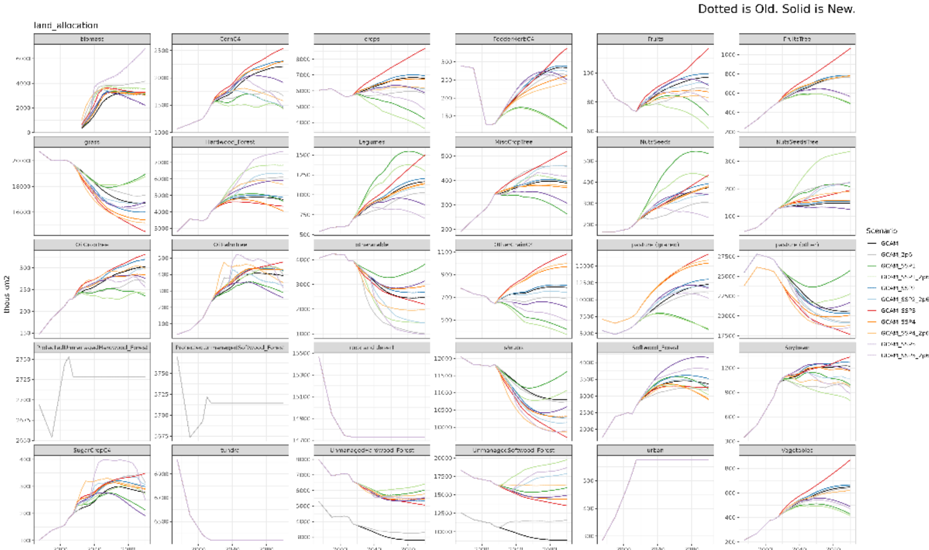
Dotted is Old. Solid is New.



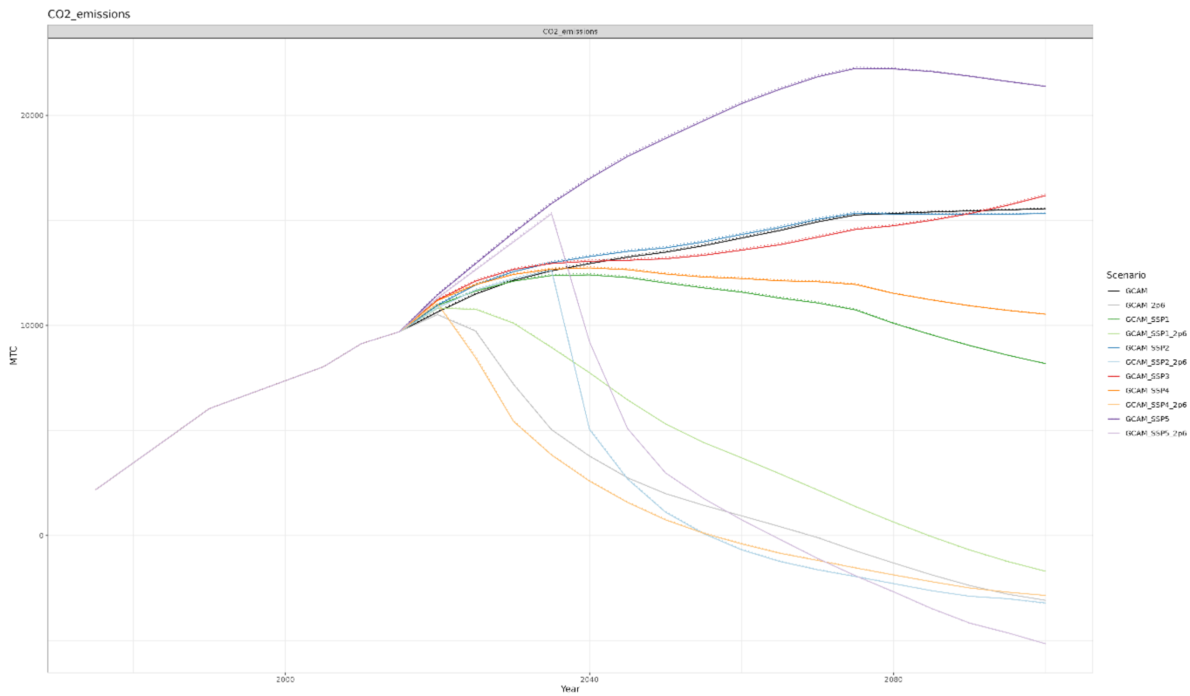
Dotted is Old. Solid is New.



Dotted is Old. Solid is New.



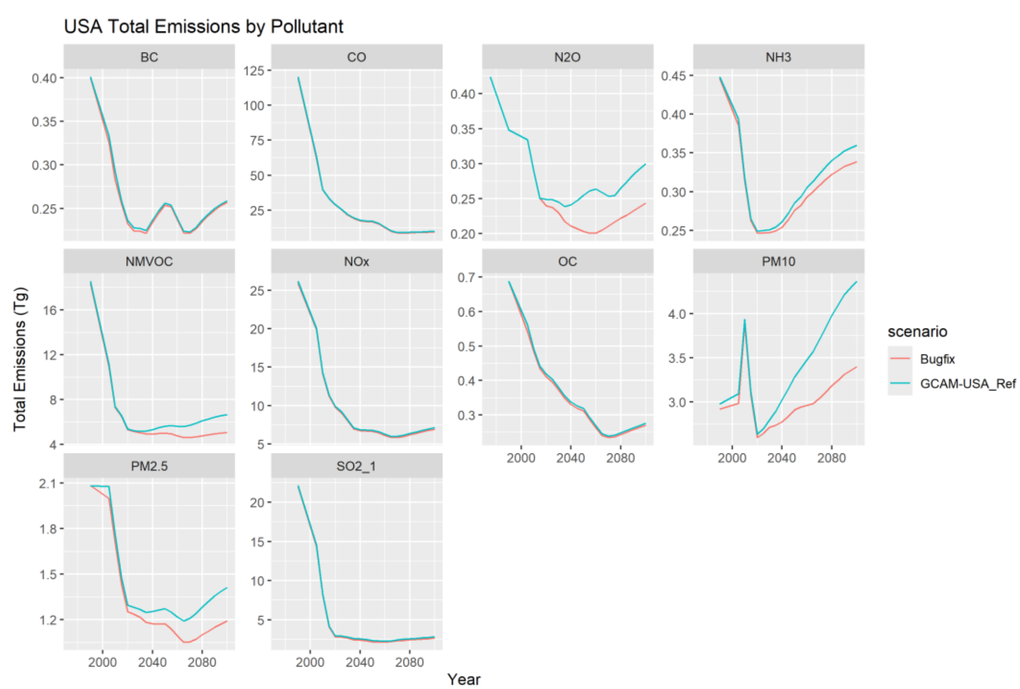
Dotted is Old. Solid is New.



Dotted is Old. Solid is New.

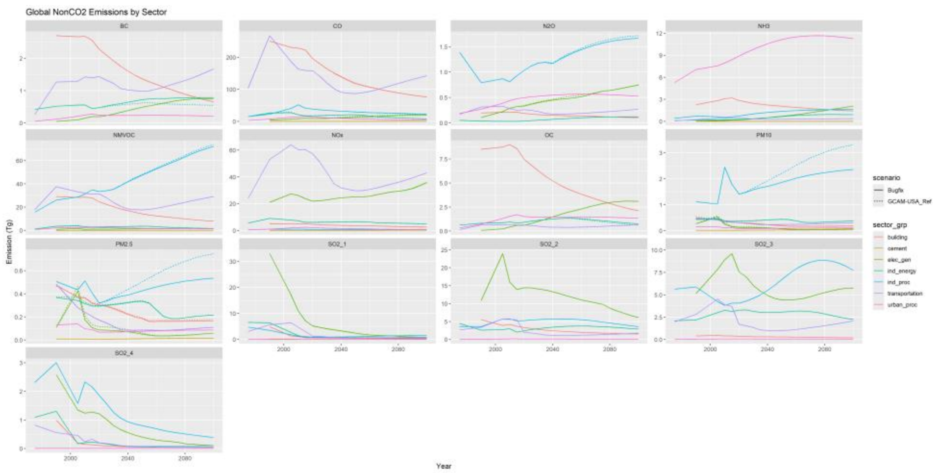
Outlier Emission Factor Replacement Function (GCAM-USA)

Non-CO2 emissions by pollutant do not vary much after the implementation of the outlier emissions factor replacement function. Changes seen in N2O emissions are largely due to the Maine coal fix, and changes to NH3, NMVOC, PM2.5, and PM10 emissions are due to the change to industrial process emissions.

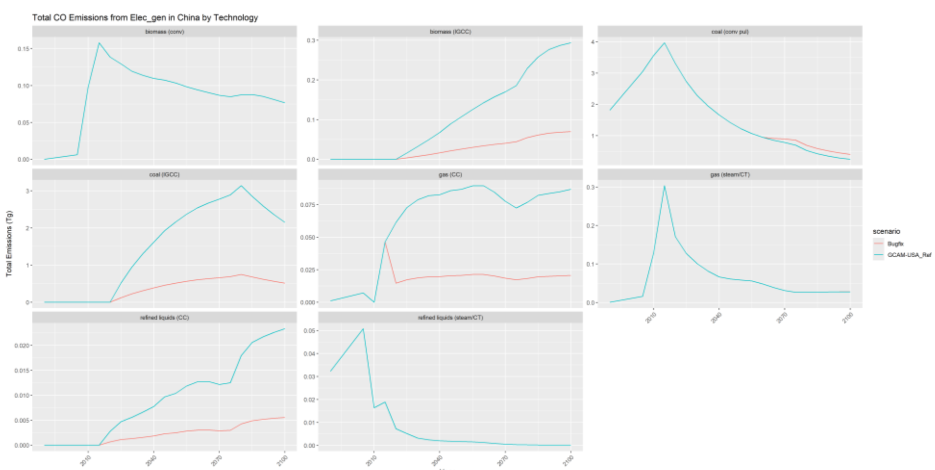


Outlier Emission Factor Replacement Function (GCAM)

Most changes shown here are driven in the USA by the fixes described above. Changes induced by the outlier replacement function include an increase in BC emissions from Buildings in Australia_NZ, a decrease in CO emissions from Electricity Generation in China, and others. A case study for one example is provided.

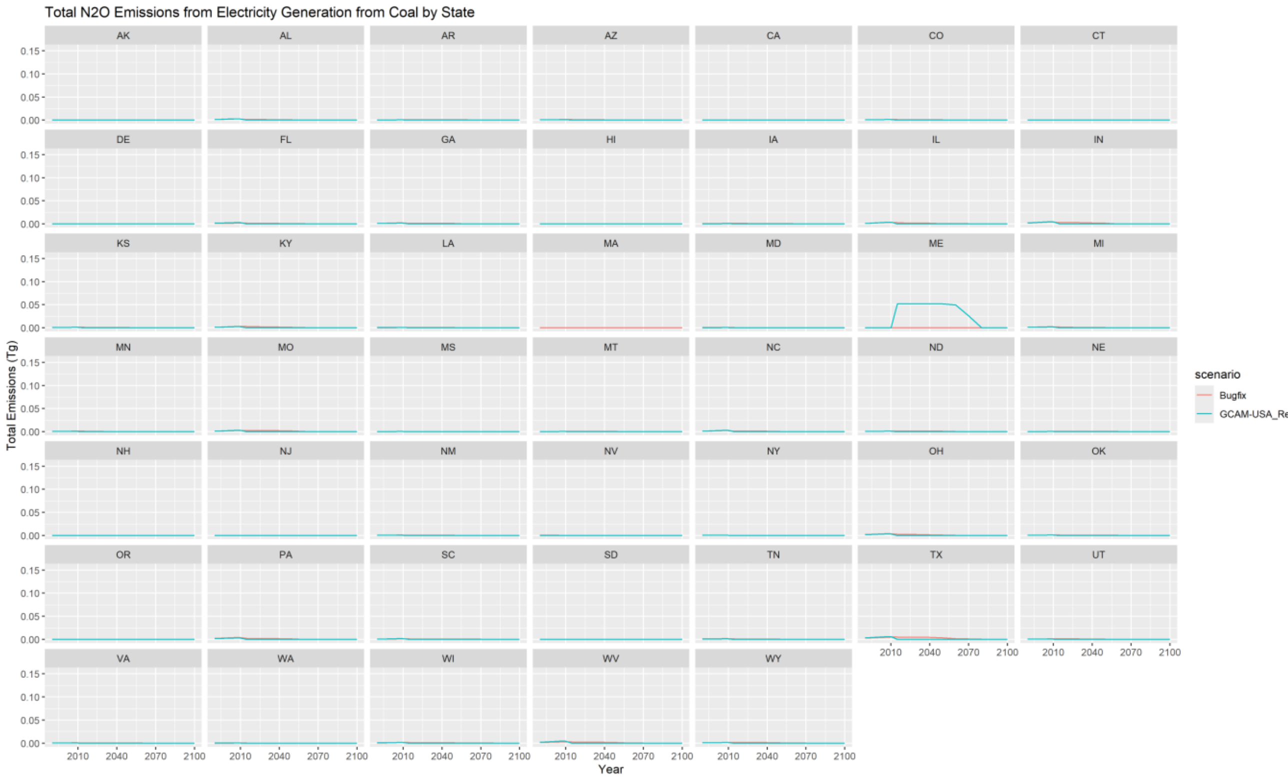


The decrease in CO emissions from electricity generation in China is driven by several technologies, most notably from biomass and coal IGCC, which come in after 2015.

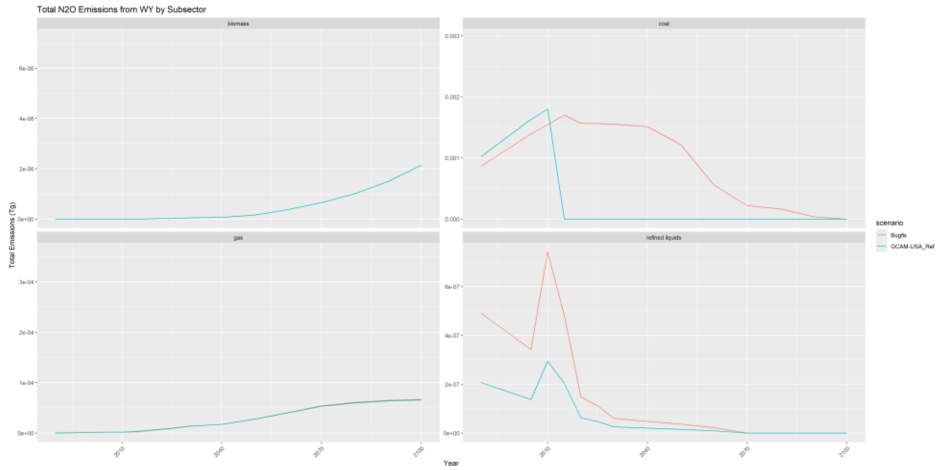


Maine N2O Emissions from Coal Electricity Generation (GCAM-USA)

These figures highlight the issue described above, where Maine was being assigned all USA N2O emissions from electricity generation from coal. The first figure is unscaled to show the magnitude, while the y-axis in the second figure varies to show the trends before and after this bugfix. A case study is provided for Wyoming.

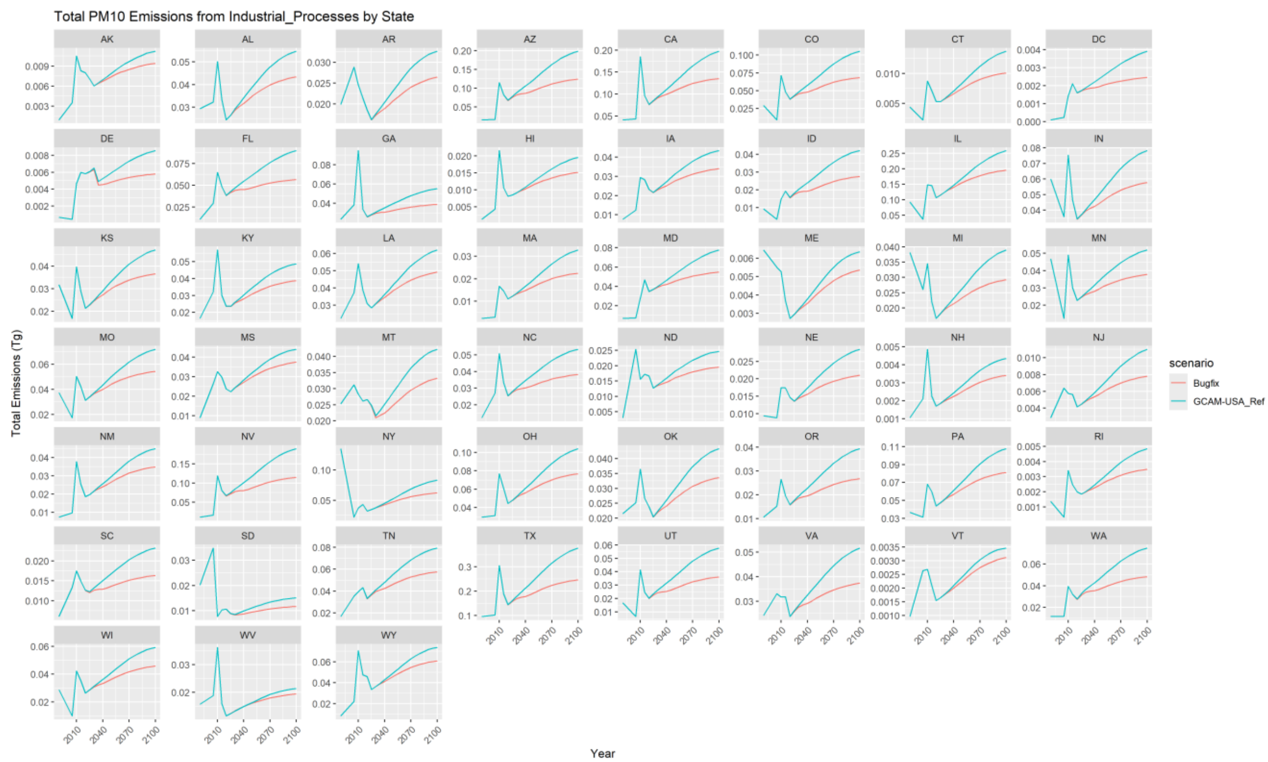


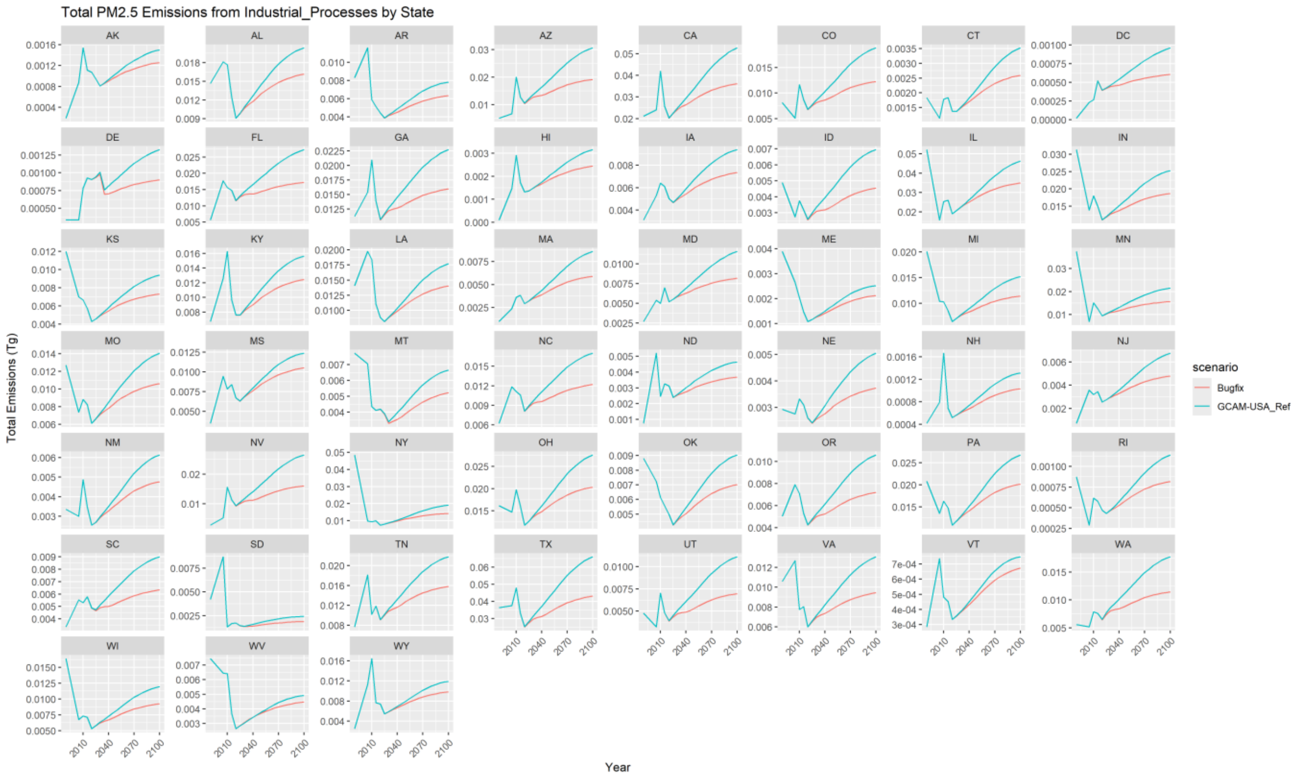
Before the bugfix, N2O emissions from electricity generation from coal dropped to 0 in 2015 for all US states except for Maine. Now, each state has emissions post-2015.



GCAM-USA Industrial Process Emissions (GCAM-USA)

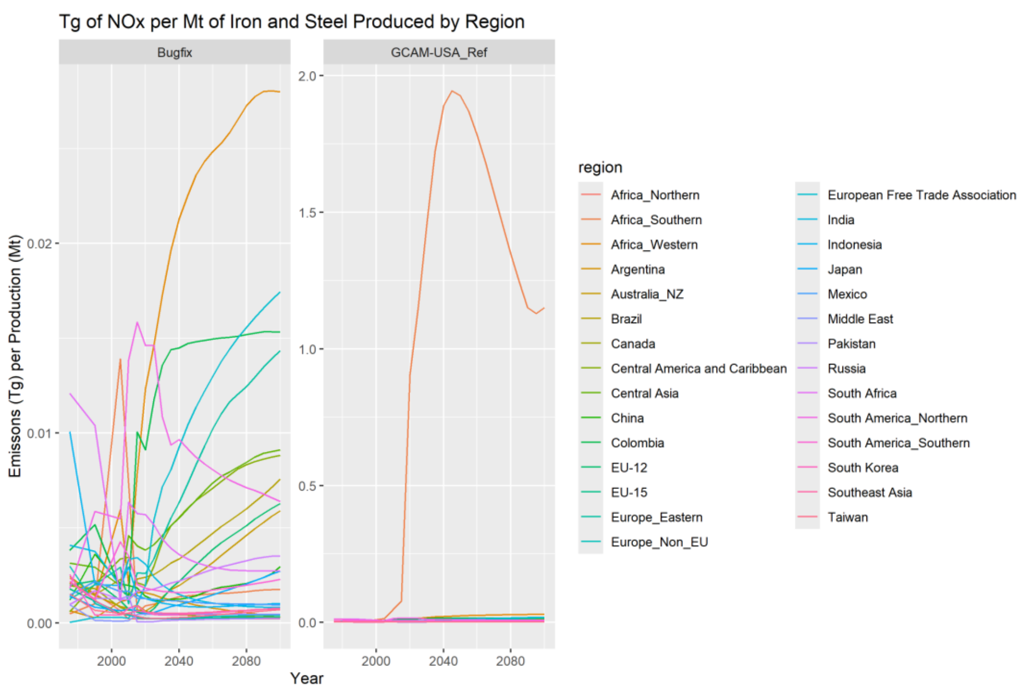
These figures display the reduction in emissions from the change to the industrial processes sector, described above. The same behavior is seen for all pollutants.





Iron and Steel (GCAM)

Previously, the Africa_Southern region had emissions factors for iron and steel emissions that were magnitudes higher than other regions. NOx emissions factors for iron and steel, by region, is shown after and before the bugfix.



These outlier emissions factors in Africa_Southern were driving global iron and steel emissions for all air pollutants. This figure shows the contribution of Africa_Southern's iron and steel emissions to global iron and steel emissions, by pollutant. For example, for NOx, global iron and steel emissions were 80Gg in 2100 (purple) and Africa_Southern contributed 75Gg of that (green). Now, global iron and steel emissions for NOx are a more reasonable 5Gg by 2100 (blue), and Africa_Southern contributed only 0.1Gg of that (red).