Core Model Proposal #384:
update to Hector v3

Product: Global Change Analysis Model (GCAM)

Institution: Joint Global Change Research Institute (JGCRI)

Authors: Pralit Patel, Kalyn Dorheim

Reviewers: Steven J Smith, Haewon C McJeon

Date committed: 04 June 2023

IR document number: PNNL-34476

Related sector: climate

Type of development: feature

Purpose: The purpose of this proposal is to update the version of Hector used in GCAM to Hector v3. More specifically:

1. Provide new hector-gcam ini and emission files
2. Add the ability to report CO2 emissions in positive terms (FFI vs DACCS) as now required by Hector v3
3. Update the emissions variable name mapping between Hector and GCAM
4. Update C++ language standard and drop Boost System/Filesystem accordingly
5. Make available more definitions of Global Mean Temperature.

Description of Changes

Hector v3 represents all the Hector model development that has taken place since 2019 (172 merged PRs, 451 files changed, 7 contributors), these changes include the addition of new features, several science upgrades/changes, new documentation, software upgrades, new default parameterization, and new historical emission pathways. However, not all these changes are necessarily relevant to the Hector-GCAM pipeline. Therefore, this core model proposal will focus only on the Hector-GCAM relevant changes. See the Hector V3 documentation manuscript for a more complete discussion of the Hector V3 changes (in prep; Pressburger 2023).

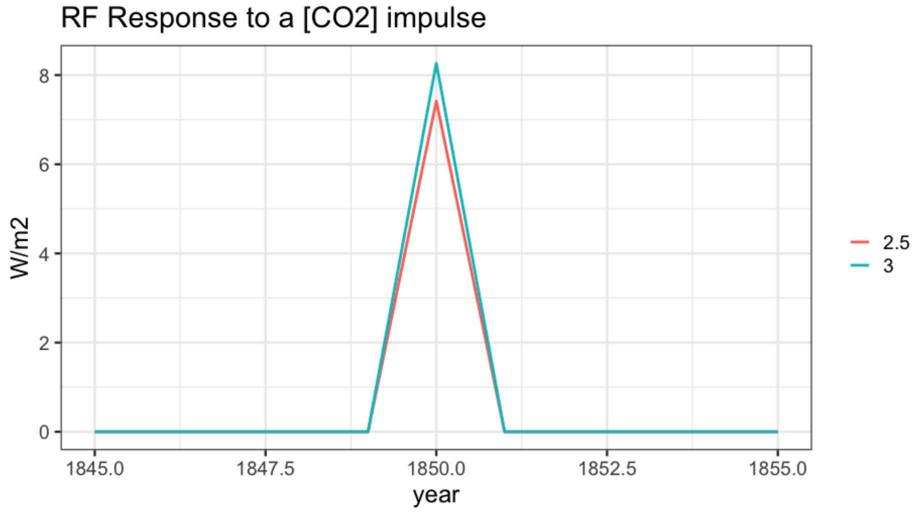
New emission pathways

Hector V3 now has the ability to account for the forcing effects of NH3 emissions, this has implications on Hector's total radiative forcing and Hector's coupling with GCAM (NH3 is now passed from GCAM to Hector).

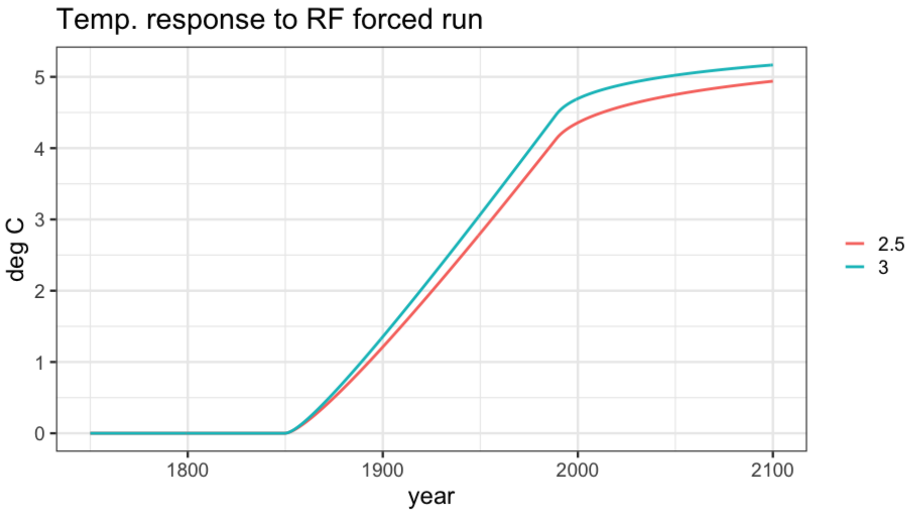
Hector V3 splits net CO2 emissions into emissions and uptake. Fossil fuel and industry emissions and LUC emissions cannot be negative. Now instead of negative emissions, negative fossil fuel and industry (FFI) emissions and land use change (LUC) emissions are classified as direct air capture with carbon storage (DACCS) uptake and LUC uptake, respectively. This has implications for Hector's coupling with GCAM. In addition, Hector would prefer to have **gross** positive and negative emissions. The mechanics of how this was done in GCAM is detailed below in "Changes in GCAM to accommodate Hector v3".

Changes to Hector's Climate System

All the equations used to calculate the radiative forcing of a particular radiative forcing agent were updated to be consistent with AR6. This means that Hector V2.5 and V3 have a different relationship between a unit of emission of X and radiative forcing. **Figure 1** shows how Hector V3 has a stronger forcing response to a change in CO2 concentrations. Meaning that Hector's climate is going to be more sensitive to changes in atmospheric concentrations.



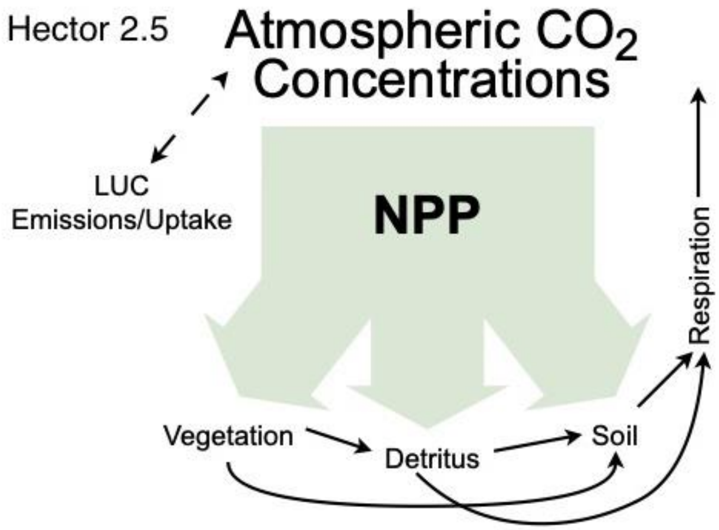
In a similar vein, changes to Hector's temperature component (aka how DOECLIM the energy balance model) is coupled with other Hector components and parameterized, which means Hector V3 is a warmer model. **Figure 2** compares Hector V3 and V2.5 temperature responses to 1pct CO2 radiative forcing pathway (RF-constrained run, the total RF from Hector's 1pct CO2 submission to RCMIP I was passed into Hector. With the RF-constraint turned on Hector uses this user-provided total RF to calculate the global mean temperature; see the Constraints section of the Hector manual for details on how this works). In the year 2100 Hector V3 is 0.23 deg C warmer than V2.5 results, Hector V3 2100 temp is 5.16 deg C whereas Hector V2.5 2100 temp is 4.93 deg C. Over the course of the entire simulation period the difference between Hector V3 and V2.5 ranges from 0.0004102 to 0.3442812 deg C.

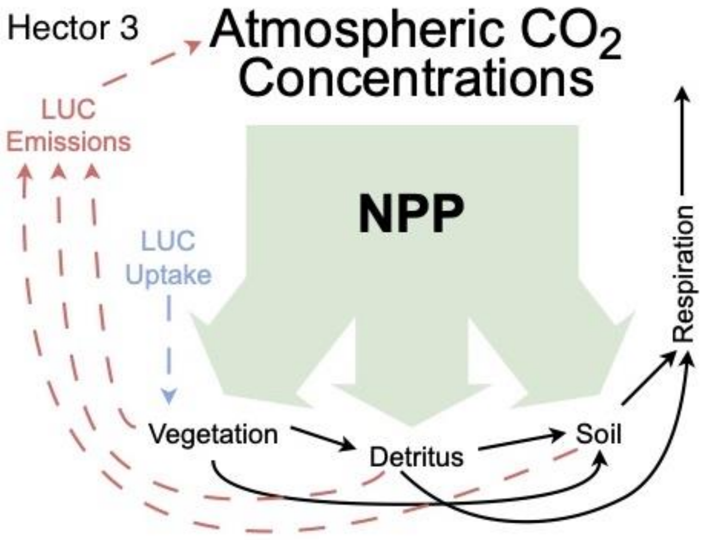


The main takeaway here is that Hector V3 is more sensitive to CO2 emissions and runs warmer than Hector V2.5.

Changes to Hector's Carbon Cycle

There was a major structural change that was made to Hector's terrestrial carbon cycle. See the two different conceptual diagrams below.





NPP is calculated as the product of a user-specified pre-industrial value ($NPP0$, default is 56.2 Pg C/year), a CO2 fertilization multiplier ($f(Catm,\beta)$), and the effect of land use change (LUC) on vegetation carbon stocks:

$NPPi(t)=NPP0,i\times f(Catm,\beta i)\times f(LUCv)$

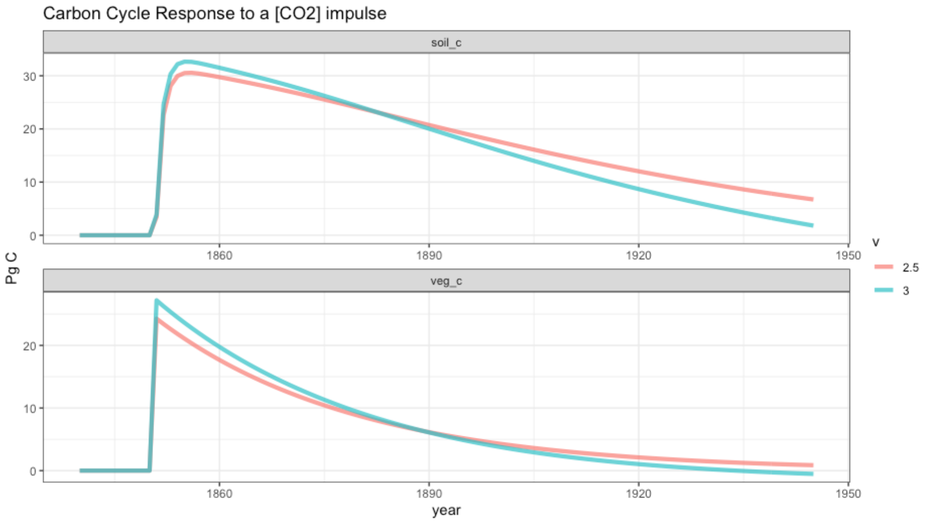
The CO2 fertilization multiplier is a function of the current atmospheric CO2 concentration ($Catm$), the initial ("pre-industrial") CO2 concentration ($C0$), and biome-specific CO2 fertilization parameter βi (default = 0.36):

$f(Catm,\beta i)=1+\beta i(\log(CatmC0))$

The LUC multiplier accounts for the effect of vegetation losses or gains on NPP (although we recognize that LUC does not always result in ‘lost’ NPP); this behavior is new as of Hector v3. It is computed as a fraction based on preindustrial (end-of-spinup) vegetation carbon (Cv) and the running sum of LUC-driven losses from – or gains to – vegetation:

$f(LUCv)=C0v-\sum LUCvegC0v$

Due to the changes in the structure of the terrestrial carbon cycle, we have different dynamics in the terrestrial carbon pools (**Figure 3**).



Now that NPP and Veg C change in response to LUC emissions in Hector V3 under scenarios with large amounts of land use change emissions, there would be higher [CO2] concentrations and weaker land carbon sink.

One of the V3 features is the NBP constraint. With the NBP constraint turned on, Hector’s land-atmosphere carbon flux is ignored and instead, Hector uses the fluxes computed by GCAM. This feature will be quite useful for future GCIMS development where GCAM and Hector carbon cycles are merged.

Hector V3 is a warmer more sensitive model, higher carbon prices may be necessary to meet climate targets.

Changes in GCAM to accommodate Hector v3

Mechanically, changing Hector versions is done by updating the Git "submodule pointer" to point to the appropriate commit of hector we now want to use. However, some additional changes were needed given changes in Hector itself.

Updated Gasses, Mappings, and Configuration

The following were updated:

- Update the `hector-gcam.ini` config file to be consistent with the latest. Also clean up the related emissions CSVs to make it more clear which emissions are provided for history (and subsequently by GCAM for "future" years) and which are gasses not provided at all by GCAM.
- Updates to `HectorModel` to match updated Hector variable names / definitions.
- Updates in `World::setEmissions` to pass NH3, directly pass HFC 23 and 32 as distinct gasses. As well as the partitioned positive / negative FFI and LUC (more details on that below).

Partitioning Emissions and Uptake

In order to partition positive and negative FFI and LUC, we make the following changes in GCAM:

- Any technology that produces negative emissions (i.e., regional biomass) will add to the FFI Uptake category via the `EmissionsSummer` object. Similarly, the positive emissions technology (all the rest) will add to the FFI Emissions category.
- For LUC we decided to treat above ground and below ground separately as GCAMs formulation for the later only makes sense on net. Thus, LUC Emissions = gross positive above ground emissions + net positive below ground emissions; LUC Uptake = gross negative above ground emissions + net negative below ground emissions. We use the `LUCEmissionsSummer` object to calculate and aggregate accordingly.
    - To partition the above ground Emissions/Uptake we need to add a new variable to `ASimpleCarbonCalc` to keep track of gross positive above ground emissions. We will then use the this in conjunction with the total to back out the gross negative (which occurs in `LUCEmissionsSummer`). Note, we chose to track the positive as it is much simpler given GCAM's assumption that all emissions occur in a single year instead of being spread out over time.

Note: Net FFI and LUC emissions are still made available for compatibility with `MagiccModel`.

LUC Related Detail in the XMLDB Output

For validating the Hector and GCAM coupling we pull out emissions from a GCAM database, process and aggregate them, then feed them to a standalone Hector (in R and via command line by generating a "default_emissions.csv"). Note, we plan on making this capability available as a standalone tool however that is not quite ready yet.

Given the method for partitioning LUC emissions / uptake noted above, this implies we need to include further detail beyond the net LUC emissions. However, LUC related results can take up a significant amount of space in an already large database. In addition, to really take the post process approach we used we must report the LUC values annually (instead of every 5 years which is the currently configured default).

As a middle ground we decided to send all of the details to the XMLDB, however we will enable, by default, a filter to remove them before being written. A user who was interested in querying emissions to feed to a standalone Hector could then easily update / remove the filter (and re-configure the reporting interval to 1 year).

The following table gives a sense of the relative tradeoffs with this approach. We can see applying filters adds some computational complexity, thus increasing the time it takes to write a database (granted on slow file systems a reduction in volume of data to write could make up for the time filtering). As we are taking this approach, we also filter out some additional rarely queried LUC related results including net above and below ground emissions and above ground carbon stocks. Note: the XML DB will strip whitespace, telling the STX filter it case also ignore white space reduced the processing overhead significantly. This final configuration noted below is what will be set as the default in GCAM.

Test	Write time (min)	Size in DB (GB)
No Filters	3:07	3.5
Filter Above Positive	4:34	3.3
Filter All LUC Detail	4:17	2.8
Filter All + No Whitespace	3:30	2.8

Updates related to Hector's switch to the C++ 17 standard

Hector v3 switched to the C++ 17 standard; thus, GCAM will make the switch too (GCAM used to be on the C++ 14 standard). The motivation for the switch was that it allowed us to drop the need to compile and link the Boost System and Filesystem libraries, which helped simplify the compiling and installation of our models. Note, technically we do allow ourselves to fall back to the C++ 14 standard however users will need to have the aforementioned Boost libraries available, details are available on Hector's build documentation.

Note, the switch means older compilers platforms may no longer be supported.

Compiler / Platform	Min Version for C++ 17
GCC	7*
macOS / Clang	macOS 12 (Catalina)
Visual Studio	2017

Clang	5

For GCAM specifically this change requires:

- Updating the Makefile / project files to note the updated standard and drop the Boost library dependencies
- The class `std::auto_ptr` was deprecated in C++ 14 (it used to generate a lot of warnings) and removed in C++ 17. It was deprecated in favor of `std::unique_ptr` which is essentially the same as auto_ptr except it takes advantage of "move" language features which, amongst other things, makes them safe to store inside of std containers such as `vector` or `map`. Given GCAM's proper usage of auto_ptr the change is a simple find/replace however it will show up in a lot of .h / .cpp files.
- The Functor base class `std::binary_function` etc. have been deprecated since C++ 11 and removed in C++ 17. They became redundant; thus, our comparison Functors no longer need to inherit from it and instead just inherit from nothing.

Additional definitions of Temperature are now Available

At present GCAM has collected and reports a single definition of temperature: global mean air temperature (GMAT). In addition, this is temperature anomaly relative to a reference period defined by the climate model. Increasingly, the IPCC is using reference period of 1850-1900 for temperature anomaly and using global mean surface temperature (GMST).

Therefore, we have decided to make all combinations of these temperature definitions available both in the C++ (IClimateModel) as well as reported to the XMLDB results. **Note**: a caveat here is that the reference period adjustment should be sensitive to Hector parameter choices; however, due to technical limitations, at the moment GCAM is always adjusting using the current default set of Hector parameters.

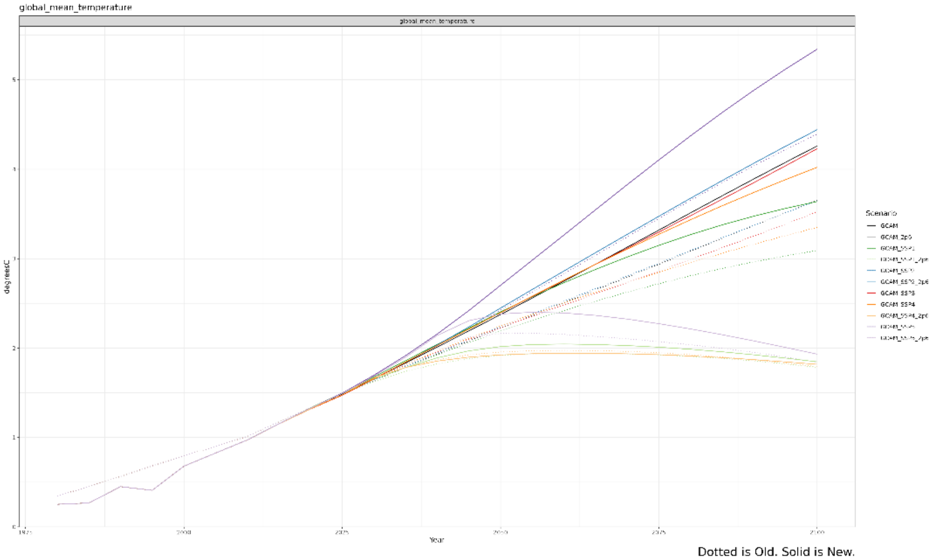
The default definition for a temperature climate target will now be GMST / 1850-1900 definition. Similarly, the "global mean temperature" query in the Main_queries.xml will by default pull GMST / 1850-1900 however comments are included noting the availability of the alternative definitions and which XML tag to query for if so interested. Also provided in the following table:

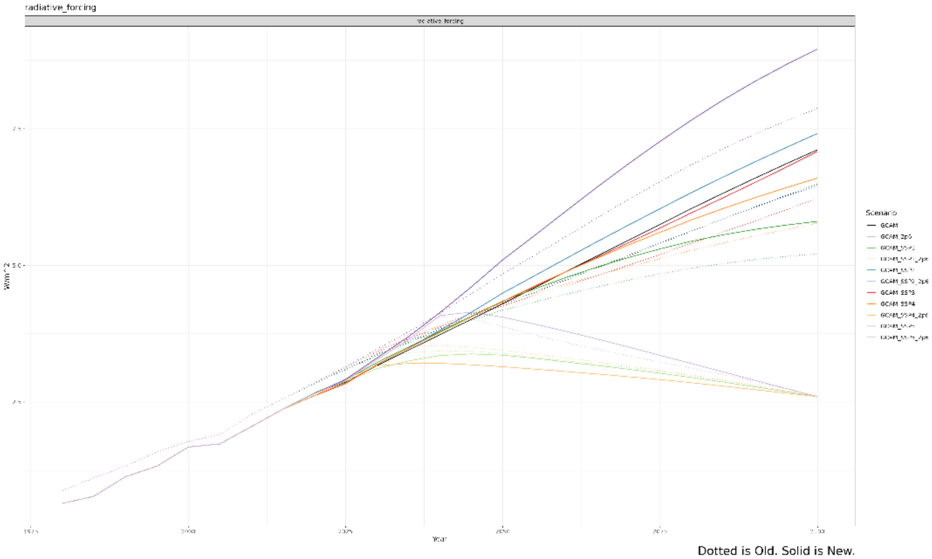
Definition	IClimateModel method	XMLDB tag
GMAT / Native	getTemperature aAdjHistoricalPeriod=false	global-mean-air-temperature-native
GMAT / 1850-1900	getTemperature aAdjHistoricalPeriod=true	global-mean-air-temperature

GMST / Native	getGmst aAdjHistoricalPeriod=false	global-mean-surface-temperature-native
GMST / 1850-1900	getGmst aAdjHistoricalPeriod=true	global-mean-surface-temperature

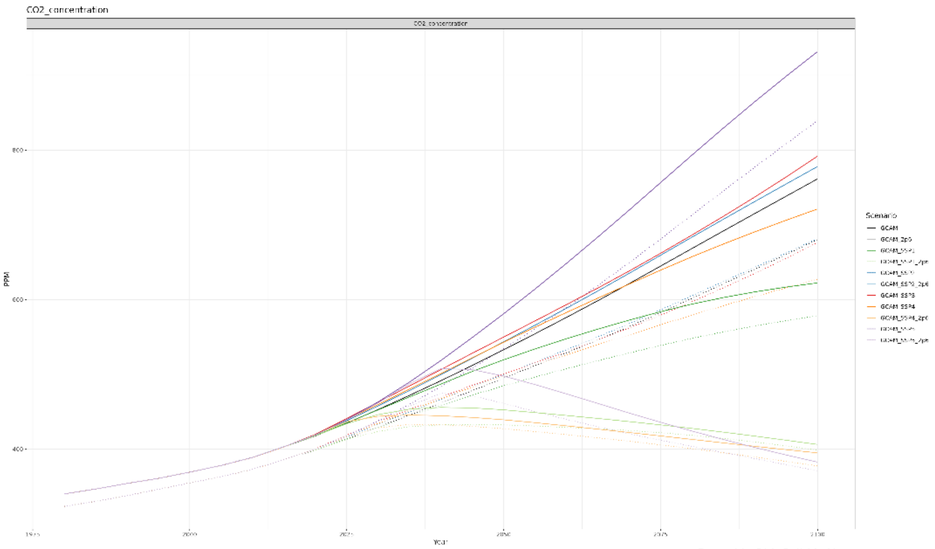
Validation

As noted earlier, Hector v3 runs hotter than Hector 2.5:



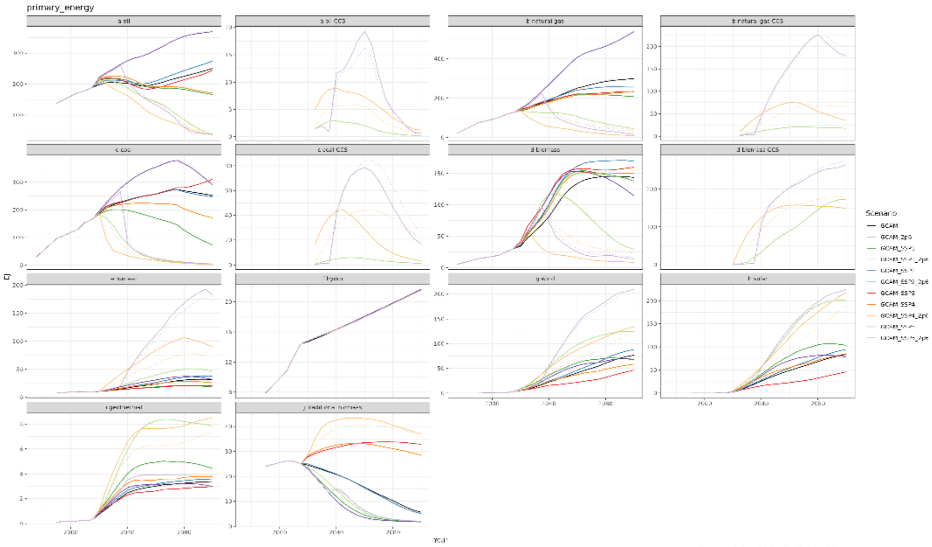


And given the improved carbon cycling also results in high CO2 concentrations in the atmosphere:

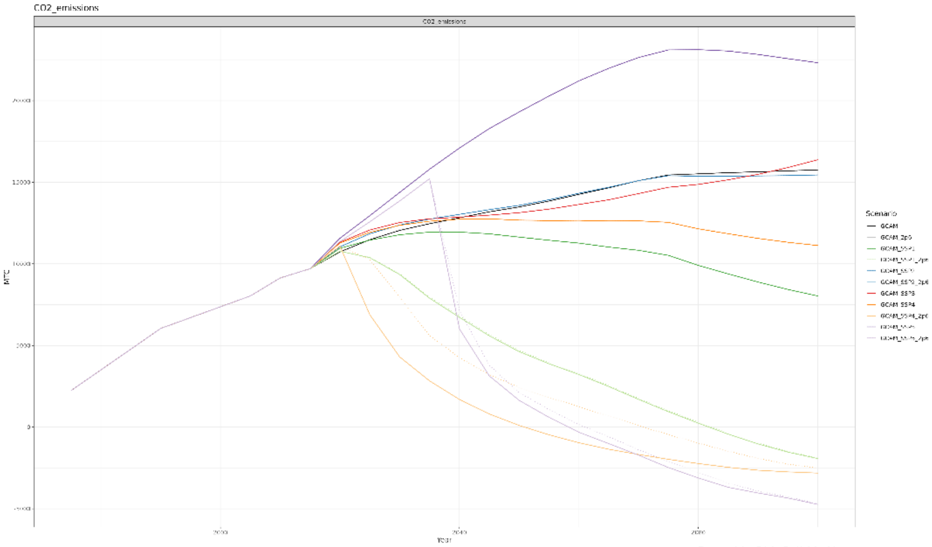


Dotted is Old. Solid is New.

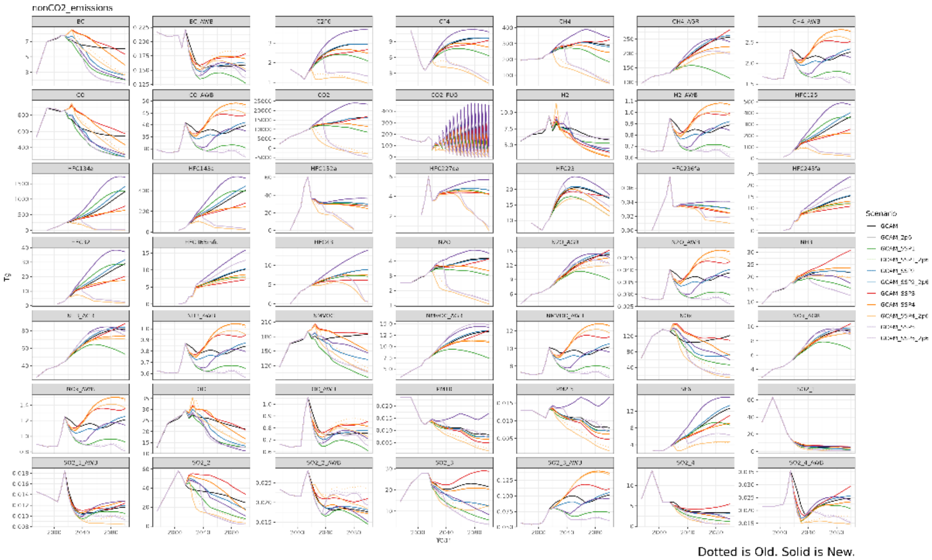
Of course, in a Reference scenario we do not expect any changes in GCAM Energy/Ag/Emissions. In policy scenarios we do as we need a higher mitigation effort from GCAM to reach the same climate target.



Dotted is Old. Solid is New.



Dotted is Old. Solid is New.



Dotted is Old. Solid is New.

References

Pressburger, Leeya, Kalyn Dorheim, Trevor Keenan, Haewon C. McJeon, Steven J. Smith, and Ben Bond-Lamberty. 2023. “Quantifying Airborne Fraction Trends and the Destination of Anthropogenic CO2 by Tracking Carbon Flows in a Simple Climate Model.” *Environmental Research Letters: ERL [Web Site]*. http://iopscience.iop.org/article/10.1088/1748-9326/acca35.