Core Model Proposal #366:
Fix Extrapolation in Residue Biomass

Product: Global Change Analysis Model (GCAM)

Institution: Joint Global Change Research Institute (JGCRI)

Authors: Pralit Patel, Xin Zhao

Reviewers: Marshall Wise, Gokul Iyer

Date committed: September 4, 2022

IR document number: PNNL-34047

Related sector: energy

Type of development: C++ code

Purpose: There are three different places in GCAM where we used the PointSetCurve to represent a piece-wise linear curve. However, the PointSetCurve has some undesirable behavior, most notably that it will extrapolate values beyond the end points of the curve. We do not actually want that behavior in these three places. To avoid that behavior, we explicitly guard against it by fixing the "output" value at the top/bottom of the curve. However, there are a couple of cases where we fail to implement these guards, most notably allowing residue biomass curves to extrapolate beyond the top of their curves. Luckily, residue biomass is parameterized up to 10 $/GJ which is a pretty high biomass price. We would not see such high prices in most GCAM scenarios, but certainly have explored such scenarios in the past.

Description of Changes

Background

The PointSetCurve is used to represent piece-wise linear curves for MAC curves, Fraction Secondary Outputs (DDGS), and Residue biomass. (Note, it is also used to calculate and store outputs of our "Policy Cost Calculation" and therefore contains **many** more bells and whistles than is needed for these three places). In each of these places, we *should* be protecting against extrapolation beyond end points of the curve at **both** the top and the bottom of the curve.

We did not do this for the top of the residue biomass curve (and the bottom of the secondary outputs but as parameterized this would only be problematic at negative prices for DDGS which we do not observe).
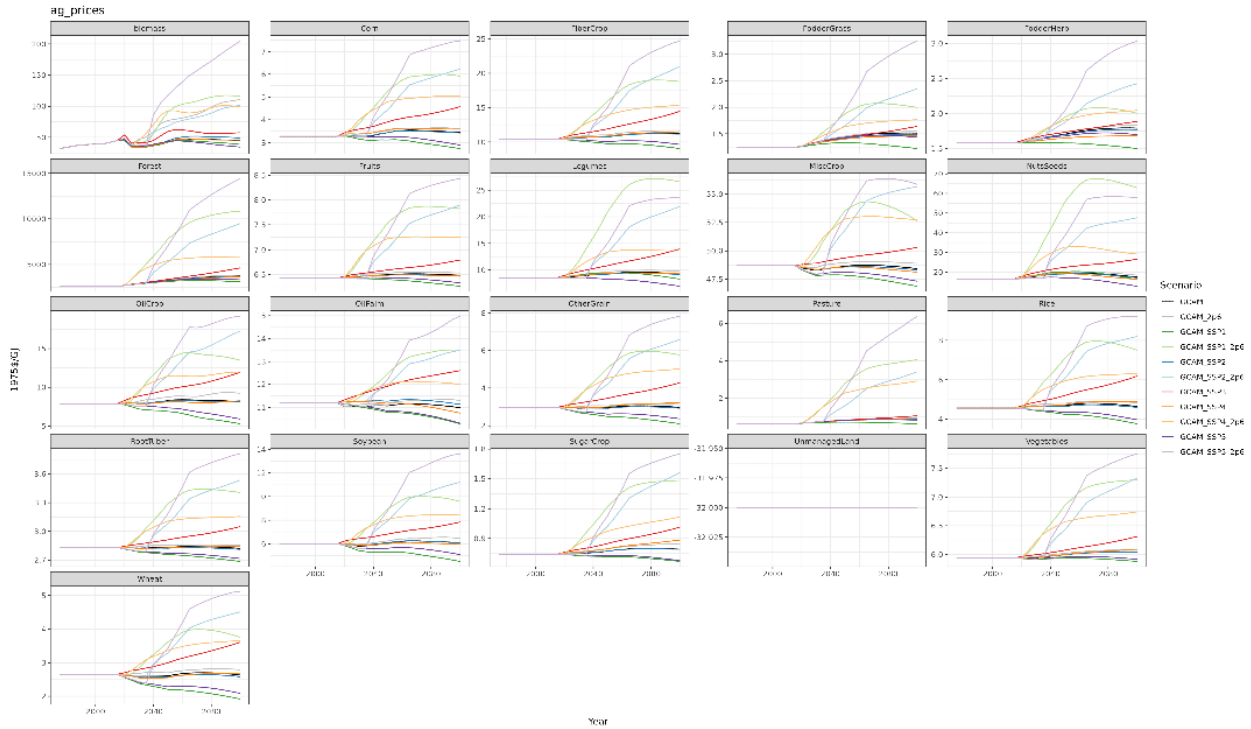
Approach

The simplest solution to this problem is to cap the value in the places where these bound checks were missing. However, having to manually implement these bound checks is error prone, so it seemed a good idea to fix the issue "upstream" so it doesn't recur. Looking into `PointSetCurve`, it is apparent the algorithm is not good for several reasons: performance wise (these curves will get looked up tens of thousands of times per iteration), extra member variables (memory usage), and of course the extrapolation issue.
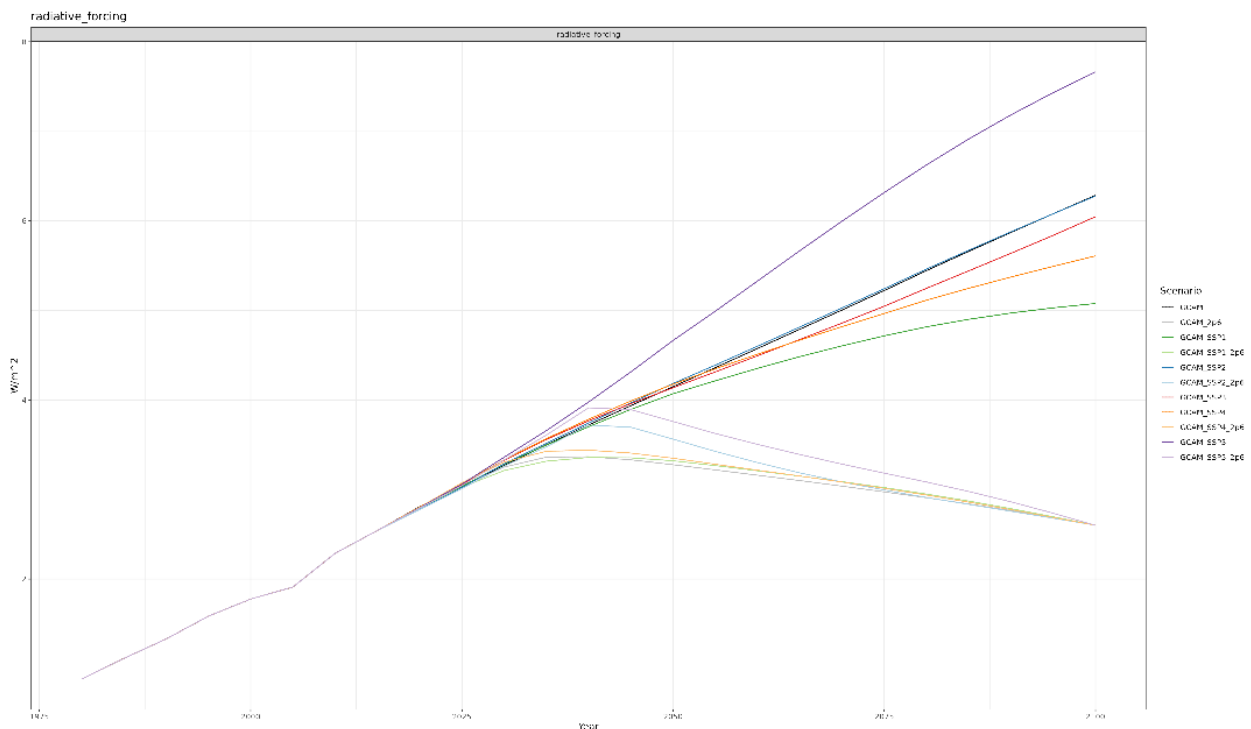
Thus, we chose to simplify and replace the usage of a `PointSetCurve` with a simple `std::map` and an associated util function `curve_lookup_interp` to do the curve lookup. This allows us to simplify the usage in each of these places and alleviates the need for explicit bound checks. This solution also simplifies the XML parsing and alleviates the performance concerns related to `PointSetCurve`.
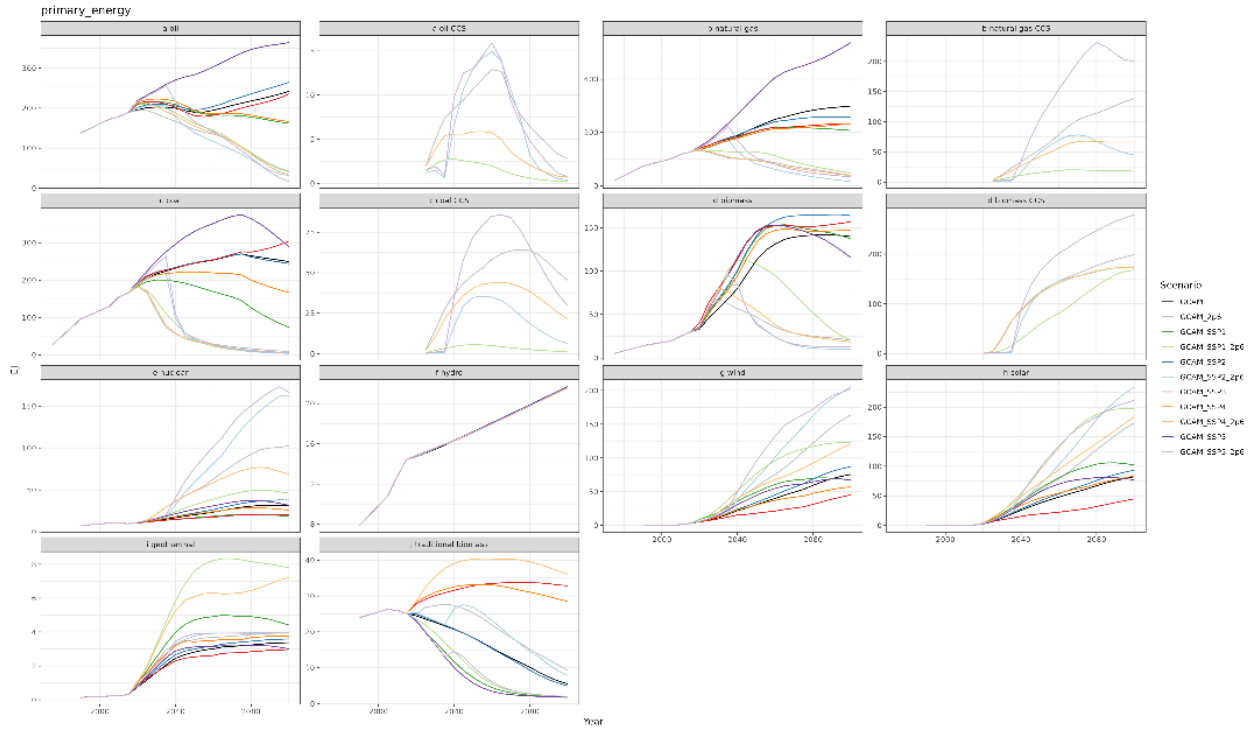
Validation

As noted, this won't affect any of the standard validation scenarios as we do not observe biomass prices > 10 $/GJ:
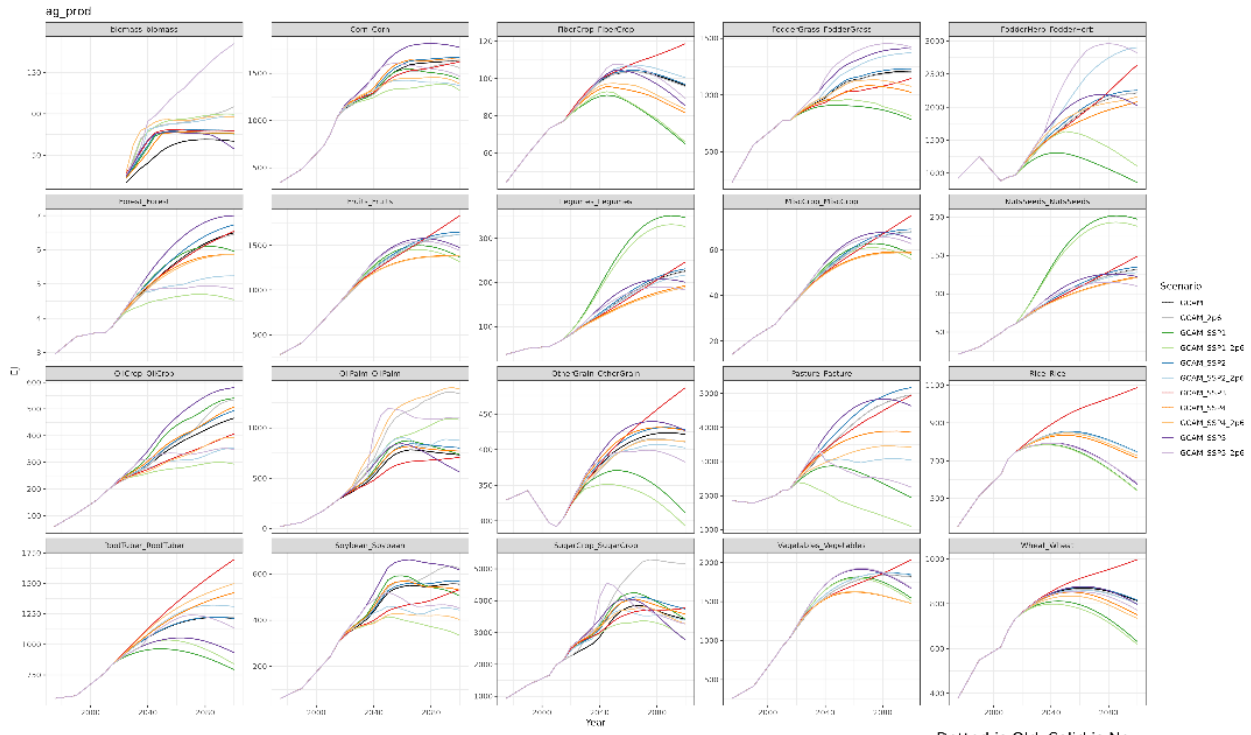
ag_prices

Dotted is Old. Solid is New.

radiative_forcing

Dotted is Old. Solid is New.

primary_energy

Dotted is Old. Solid is New.

ag_prod

Dotted is Old. Solid is New.

4

To verify that these changes work as intended we ran a scenario with:

1. Lowered land logits, limiting expansion into unmanaged eco-systems. This drives up crop and biomass prices more quickly.
2. A Global UCT scenario, and a fixed carbon tax at 100 $/tC + 5%/yr
3. No limits on biomass whatsoever.

We can see in 2070 is where the biomass price crosses $10 (for the USA in the figure; this is the case everywhere):
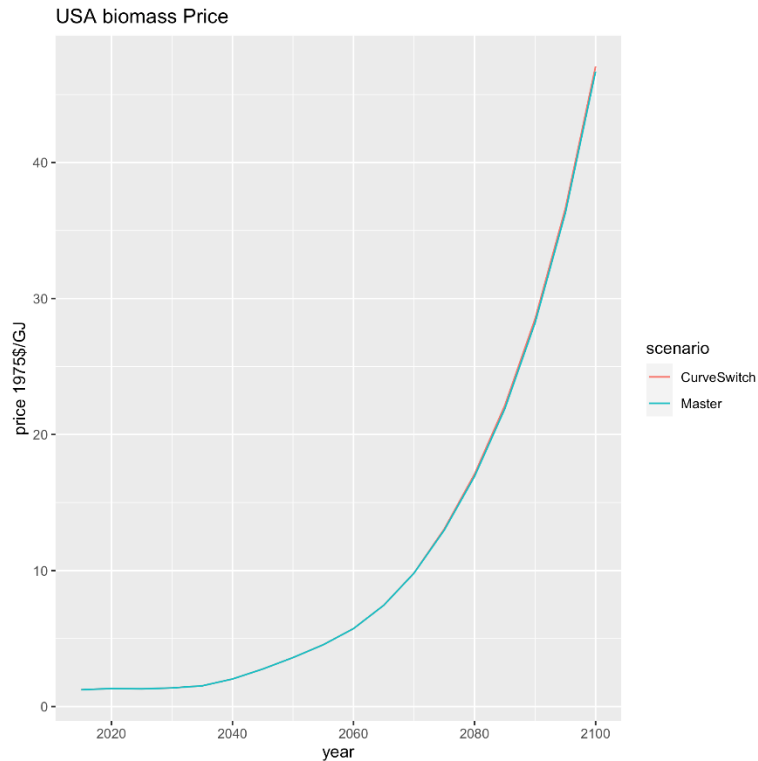
We don't see divergence in behavior until after that point. We see the behavior move as expected, with continued growth in biomass supply from residues in the Core but not so in this proposal (CurveSwitch):

Global Residue Biomass
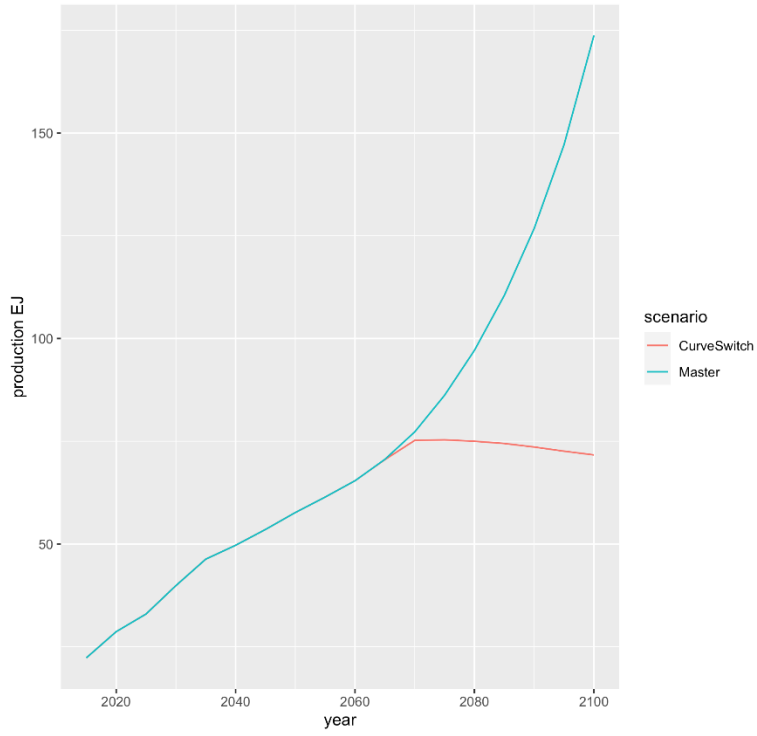
Global Total Biomass Production