Attribution of tropospheric ozone production in CAM-chem using an extended tagging technique

Tim Butler, Jane Coates, Aurelia Lupascu, and Shuai Zhu¹

Institute for Advanced Sustainability Studies, Potsdam
(tim.butler@iass-potsdam.de)

¹Now at IBM China research lab
Overview

- Existing tagging in CAM-chem
- New extensions
- Implementation
- Initial results
- Next steps
Fig. 6. Zonal average of tagged ozone source contributions at the surface and at 400 hPa, for January and July of 2008. Stratospheric contribution is determined as the difference between total ozone and tagged ozone from all tropospheric sources combined. Emmons et al. (2012)
Established tagging in CAM-Chem

- Tagged source duplicated as emissions of “XNO”
  - \( \text{NO} + \text{RO}_2 \rightarrow \text{NO}_2 + \text{RO} \)
  - \( \text{XNO} + \text{RO}_2 \rightarrow \text{XNO}_2 \) (+ \( \text{RO}_2 \))
  - \( \text{XNO}_2 + \text{hv} \rightarrow \ldots \rightarrow \text{O}_3\text{A} \)
- Tag is followed through all other reactions involving \( \text{NO}_x \) and \( \text{O}_3 \)
- One tag allowed per model run
  - Tagging multiple \( \text{NO}_x \) sources requires multiple runs
- What happens with the “null cycle”
  - \( \text{O}_3 + \text{NO} \rightarrow \text{NO}_2 \) and then ( \( \text{NO}_2 + \text{hv} \rightarrow \ldots \rightarrow \text{O}_3 \))
  - Which tag does \( \text{NO}_2 \) get?
  - It gets the tag from \( \text{NO} \)
  - The null cycle replaces \( \text{O}_3 \) tags!
Extended tagging in CAM-Chem

- Allow for multiple tags in a single model run
- Tagged sources emitted as NO_{XXX}
  - NO_{TAG} + RO_2 \rightarrow NO_2_{TAG}
  - NO_{FOO} + RO_2 \rightarrow NO_2_{FOO}
  - NO_{BAR} + RO_2 \rightarrow NO_2_{BAR}
- Null cycle
  - O_3 + NO \rightarrow NO_2
  - Possibility for NO_2 to inherit from both O_3 and NO_2
Extended tagging in CAM-Chem

- Introduce a new set of odd-oxygen tracers “X_TAG” species
- \( \text{NO} + \text{RO}_2 \rightarrow \text{NO}_2 + \text{RO} \)
- \( \text{NO}_{\text{TAG}} + \text{RO}_2 \rightarrow \text{NO}_2_{\text{TAG}} + \text{NO}_2_{\text{X_TAG}} \)

  + \( \text{NO}_2_{\text{TAG}} + \text{hv} \rightarrow \text{NO}_{\text{TAG}} \)
  + \( \text{NO}_2_{\text{X_TAG}} + \text{hv} \rightarrow \ldots \rightarrow \text{O}_3_{\text{X_TAG}} \)

- Null cycle
  - \( \text{O}_3 + \text{NO} \rightarrow \text{NO}_2 \)
  - \( \text{O}_3 + \text{NO}_{\text{TAG}} \rightarrow \text{NO}_2_{\text{TAG}} \)
  - \( \text{O}_3_{\text{X_TAG}} + \text{NO} \rightarrow \text{NO}_2_{\text{X_TAG}} \)
Implementation

- Perl Scripts
  - Specify a list of tags to apply (e.g. “TAG”, “FOO”, “BAR”)
  - Rewrite the mechanism file, adding new reactions and tracers
  - Rewrite model source code (tracers, reaction rates, etc…)

- Each tag adds:
  - 155 reactions
  - 26 tracers
Model setup

- CESM 1.2.2, FSDCHM compset
- Modified mechanism from Emmons et al. (2012)
- Modified chemical preprocessor
- HTAP2 experiment protocol
Evaluation: comparison with the HTAP2 ensemble and CAM-chem
Comparison with CAM-chem HTAP2 run

Annual average surface ozone
North America: Monthly Average O3 for Emissions and Meteorology for Year 2010

Total Ozone Mean (ppbv) and Mean from NOx Sources in Tier 2 Receptor Regions

Monthly Mean O3 of Transported NOx Sources from Source Regions to Tier 2 Receptor Regions

Percent Contributions of Transported NOx Sources from Source Regions to Tier 2 Receptor Regions
Summary

- Extended version of existing tagging in CAM-chem
- Support for multiple tags
- $O_x$ cycling preserves tags
- Application to HTAP2 source/receptor modelling

Next steps:
- Obtain a tagged simulation consistent with a scientifically supported CESM compset
  - Most appropriate configuration of CESM2?
- Add VOC tagging (already in testing)
Implementation

• Original code:

```fortran
if( jno2a_ndx > 0 .and. jno2_ndx > 0 ) then
    photos(:,:,jno2a_ndx) = photos(:,:,jno2_ndx)
end if
```

• Manually modified “template” code:

```fortran
! BEGIN TAGGING CODE
if( jno2_tag_ndx > 0 .and. jno2_ndx > 0 ) then
    photos(:,:,jno2_tag_ndx) = photos(:,:,jno2_ndx)
end if
! END TAGGING CODE
```

• Template code is then automatically processed to produce compiler-ready code...