Community Terrestrial Systems Model (CTSM) Update and
CTSM5.1 (CLMBGC) Perturbed Parameter Experiment

David Lawrence, Bill Sacks, Negin Sobhani, Sam Levis, Mariana Vertenstein, Mike Barlage, Fei Chen, Martyn Clark, Erik Kluzek, Keith Oleson, Daniel Kennedy, Katie Dagon, Forrest Hoffman, Rosie Fisher, Sean Swenson, Ben Sanderson, Nate Collier, Will Wieder, Danica Lombardozzi, Gordon Bonan, ...
The Community Terrestrial System Model
a unified model for research and prediction in **climate**, **weather**, **water**, and **ecosystems**

- **Land-Atmosphere Interactions**
- **Climate Change**
- **Weather and Predictability**
- **Land Management**
- **Hydrology**
- **Ecology**
- **Biogeo-Chemistry**
- **Cryosphere**

**CTSM**

**CESM or any CMEPS compatible model**

- **SIMA, WRF, MPAS, or other atm or data model**

**CIME**

**LILAC**
Light-weight Infrastructure for Land-Atmosphere Coupling
WRF-CTSM coupling via LILAC

Released to community in September, 2020
User’s guide: Accessible from CTSM github wiki page
escomp.github.io/ctsm-docs/versions/master/html/lilac/specific-atm-models/wrf.html
WRF Test Simulations (27km): Spectral nudged runs

Tmax bias July 2013

WRF-Noah

WRF-NoahMP

WRF-CTSM5(NWP)

WRF-CTSM5(CLMSP)

Note: WRF-CTSM5(NWP) ~15-20% more expensive than WRF-Noah
WRF Simulations (27km)

WRF-CTSM5 (CLMSP) performs as well or better in all months in this test.
WRF Simulations (27km)

WRF-CTSM5(CLMSP) performs as well or better in all months in this test

But, it IS more expensive: WRF-CTSM5(NWP) is 5% slower than WRF-NoahMP
WRF-CTSM5(CLM-SP) is 40% slower than WRF-NoahMP
CTSM: Refactor of surface dataset generation toolchain

Benefits:

- Two clear steps for creating the surface dataset
- User has option to stop at making namelist, mapping files, or sfc dataset
- User need not be aware of intermediate (a) mapping files, (b) fortran namelist, (c) landuse.txt file
- No need for separate SRC mesh file paths; script chooses appropriate path for each raw dataset

**gen_user_namelist.py**
Using command line arguments the user sets options and this code creates a user-friendly namelist of all input raw datasets

**DST grid file**

**Raw datasets**

**Metadata**

**mkmapdata.py**
- Reads namelist for DST mesh file
- For each raw dataset in namelist, finds SRC mesh file and mask from the netcdf metadata
- Checks if the weight (mapping file) already exists or not (creates if not)
- If it does not exist, it creates the mapping

**Metadata from raw datasets include SRC grid files**

**Surface dataset**

**Benefits:**

- Two clear steps for creating the surface dataset
- User has option to stop at making namelist, mapping files, or sfc dataset
- User need not be aware of intermediate (a) mapping files, (b) fortran namelist, (c) landuse.txt file
- No need for separate SRC mesh file paths; script chooses appropriate path for each raw dataset

**mkmapdata.py**

**mksurndata_map (Fortran)**

**Mapping files**

**namelist**

User can modify

**Raw datasets**

**Metadata**

**DST grid file**

**Benefits:**

- Two clear steps for creating the surface dataset
- User has option to stop at making namelist, mapping files, or sfc dataset
- User need not be aware of intermediate (a) mapping files, (b) fortran namelist, (c) landuse.txt file
- No need for separate SRC mesh file paths; script chooses appropriate path for each raw dataset

**mkmapdata.py**
- Reads namelist for DST mesh file
- For each raw dataset in namelist, finds SRC mesh file and mask from the netcdf metadata
- Checks if the weight (mapping file) already exists or not (creates if not)
- If it does not exist, it creates the mapping

**Surface dataset**

**Benefits:**

- Two clear steps for creating the surface dataset
- User has option to stop at making namelist, mapping files, or sfc dataset
- User need not be aware of intermediate (a) mapping files, (b) fortran namelist, (c) landuse.txt file
- No need for separate SRC mesh file paths; script chooses appropriate path for each raw dataset

**mkmapdata.py**
- Reads namelist for DST mesh file
- For each raw dataset in namelist, finds SRC mesh file and mask from the netcdf metadata
- Checks if the weight (mapping file) already exists or not (creates if not)
- If it does not exist, it creates the mapping

**Surface dataset**
The Community Terrestrial Systems Model

Next steps (not necessarily in order)

- CTSM5.1 and 5.2 ‘physics’ development (see Will’s talk)
- CTSM overview paper, including rationale, description of capabilities, and WRF-CTSM assessment
- Finalize surface dataset creation workflow
- Initialization of BGC configurations for regional high-res models
- Evaluation in high-resolution (~3km) forecast mode
- Finish hydrology refactor
- LILAC coupling to COSMO, MPAS, LIS, and other regional models
- Distributed hydrology from NoahMP, WRF-Hydro
The CTSM5.1 (CLMBGC) Perturbed Parameter Ensemble Project

Dave Lawrence, Katie Dagon, Daniel Kennedy, Keith Oleson, Rosie Fisher, Forrest Hoffman, Ben Sanderson, Charlie Koven, Erik Kluzek, Danica Lombardozzi, Nate Collier, Will Wieder, Gordon Bonan, and …
CTSM5.1 (CLMBGC) Parameter Perturbation Ensemble (PPE)

Goals:

• Complete comprehensive parameter uncertainty assessment and calibration of full CLM5.1BGC model
• Develop infrastructure for easy PPEs and global parameter estimation
• Explore sensitivity of a range of features of global coupled land system to reasonable uncertainty in model parameter values

Land carbon cycle uncertainty

Bonan and Doney, 2018

Water use efficiency trends: Structural uncertainty

Uncertainty due to parameter uncertainty?

Bonan and Doney, 2018

Lawrence et al, 2019
Phase 0: Infrastructure development

Phase 1: One-at-a-time parameter ensembles under range of environmental perturbations

Phase 2: Latin-hypercube ensemble with most ‘important’ parameters

Use neural network to develop emulator of CLM output

Phase 3: Identify optimized parameter sets based on multi-objective calibration targets

Run 200-member ensemble of global transient simulations with reasonable parameter sets
Identified ‘all’ CLM5 parameters (>200)
Extract hard coded parameters to input parameter file (>100 parameters moved to parameter file)
Catalog all parameters and reasonable ranges in ‘living’ document
Phase 0: Infrastructure development - Parameters

- Identify ‘all’ CLM5 parameters (>200)
- Extract hard coded parameters to input parameter file (>100 parameters moved to parameter file)
- Catalog all parameters and reasonable ranges in ‘living’ document

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>min</th>
<th>max</th>
<th>comments?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photosynthetic capacity (LUNA)</td>
<td></td>
<td></td>
<td>feel free to add any comments below <strong>ok, to write XXpercent, in lieu of absolute range</strong></td>
<td></td>
</tr>
<tr>
<td>slatop</td>
<td>pft</td>
<td>pft</td>
<td></td>
<td>specific leaf area at the canopy top</td>
</tr>
<tr>
<td>jmaxb0</td>
<td>0.01</td>
<td>0.05</td>
<td></td>
<td>the baseline proportion of nitrogen allocated for electron transport (J)</td>
</tr>
<tr>
<td>jmaxb1</td>
<td>0.05</td>
<td>0.25</td>
<td>This is Jmaxb1 in the code (note the capital J)</td>
<td>the baseline proportion of nitrogen allocated for electron transport (J)</td>
</tr>
<tr>
<td>Plant hydraulics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kmax</td>
<td>pft</td>
<td>pft</td>
<td>see <a href="https://github.com/ESCOMP/CTSM/issues/1162">https://github.com/ESCOMP/CTSM/issues/1162</a> for how I chose kmax/kmax</td>
<td>Plant segment max conductance</td>
</tr>
</tbody>
</table>
Cluster Analysis on transient simulation, assessed on mean and interannual s.d. for ~20 forcing and carbon, water, and energy state and flux variables

- With about 300-400 clusters, can reasonably replicate 2°C global mean and transient model output
- Fast and cheap: 4 pe-hrs/yr
- Fast spinup: w/ CN Matrix (Lu et al., JAMES, 2020), full C/N spinup in ~120 years
- 1million pe-hrs = ~2000 parameter perturbation simulations, incl. spinup

**Use ILAMB to assess reconstructed output against 2°C simulation ‘truth’**
Automated scripts to:

- Setup cases
- Manipulate parameter values
- Execute and check spinup
- Conduct ensembles

Scripts are generalizable enough for other CIME-based model components (e.g., components of CESM, SIMA, etc)

Analysis scripts using Jupyter notebooks in development to reduce barriers-to-entry for exploration and analysis by multiple collaborators
Phase 1: One-at-a-time parameter sensitivity

Ensemble of one-at-a-time low/high param value simulations

- Each simulation checked for reasonableness
  - Plant survivability rate within 30% control, reasonable max LAI
  - GPP, LH within +/-30% of observed (ILAMB)
- If run with particular parameter value doesn’t pass checks, constrict parameter range and run again
Phase 1: One-at-a-time parameter sensitivity

Parallel ensembles with environmental perturbations

- Climate: 1850 and SSP3-7 CESM2 climate
- CO2: 1850 and SSP3-7
- N-dep: +5 gN/m²/yr
- Last Glacial Maximum conditions
- Restrict parameter ranges again if low-side environmental perturbation doesn’t pass reasonableness checks
Phase 2: Latin-hypercube ensemble

1. Select ~50 ‘most important’ parameters
   • Following Dagon et al., 2020, ‘objectively’ select parameters that have most significant impact on range of key carbon, water, energy flux and state variables for
     • Mean state and variability
     • Non-overlapping spatial patterns
     • Response to environmental perturbations

2. Run sparse grid simulations with ~2500 Latin hypercube-defined parameter sets
   • Present-day climate (1.5 million pe-hrs)
   • Environmental perturbations
Phase 3: Global transient 2° simulations

- With Phase 2 Latin Hypercube ensemble output, use neural network to develop emulator of CLM5 output based on parameter settings (Dagon et al., in review)
- Select ~200 ‘best’ parameter sets (selection criteria TBD, ILAMB?)

- Run full spinup and transient historical/projection period 2° simulations

We need people to help analyze these ensembles! Contact me if interested