Three Years of Doing Uncertainty Quantification
Ensemble Runs at LLNL Using CAM & CESM

17th Annual CESM Workshop
June 19, 2012

John Tannahill
The project.

- A 3 year LLNL Strategic Initiative LDRD project, started in October of 2009:
  - “The Advance of Uncertainty Quantification Science with Application to Climate Modeling, Inertial Confinement Fusion Design, & Stockpile Stewardship Science”

- Goal is to advance the UQ of multi-physics simulations that utilize large collections of uncertain input parameters.

- PI is Richard Klein.
Project made up of four teams.

- **Climate Team**
  - Curt Covey
  - Don Lucas
  - John Tannahill
  - Yuying Zhang

- **UQ Pipeline Team**
  - David Domyancic
  - Scott Brandon
  - Rao Nimmakayala
  - Jim McEnerney

- **Statistics Team**
  - Gardar Johanneson & others

- **Error Estimation Team**
  - Carol Woodward & others
CAM/CESM uncertain parameters of interest.

- Established ranges for each parameter.

- Namelist source files included:
  - `cloud_fraction`
  - `pkg_cldoptics`
  - `pkg_cld_sediment`
  - `cldwat`
  - `hb_diff`
  - `hk_conv`
  - `physpkg`
  - `zm_conv`
  - `shr_flux_mod`
  - `Biogeophysics1Mod`
  - `ice_shortwave`
  - `ice_therm_vertical`
  - `ice_mechred`
  - `hmix_aniso`
  - `hmix_gm`
  - `mix_submeso`
  - `tidal_mixing`
  - `vertical_mix`
  - `vmix_kpp`

<table>
<thead>
<tr>
<th>Namelist File</th>
<th>Number of Parameters</th>
<th>Started With</th>
<th>Reduced To</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm_in</td>
<td>27</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>drv_in</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ice_in</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Ind_in</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>pop2_in</td>
<td>18</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>54</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>
## All CAM/CESM UQ Studies to Date.

<table>
<thead>
<tr>
<th>Study Type</th>
<th>Number of Runs</th>
<th>Number of Simulated Years</th>
<th>Stored Size (TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAM Atmosphere Data Ocean</td>
<td>3,873</td>
<td>46,476</td>
<td>56</td>
</tr>
<tr>
<td>CAM Atmosphere Slab Ocean</td>
<td>823</td>
<td>25,968</td>
<td>47</td>
</tr>
<tr>
<td>Data Atmosphere POP2 Ocean</td>
<td>545</td>
<td>8,070</td>
<td>132</td>
</tr>
<tr>
<td>CAM Atmosphere POP2 Ocean</td>
<td>327</td>
<td>6,710</td>
<td>128</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td><strong>5,568</strong></td>
<td><strong>87,224</strong></td>
<td><strong>363</strong></td>
</tr>
</tbody>
</table>
Study particulars.

<table>
<thead>
<tr>
<th>Study Type</th>
<th>Code Version(s) Used</th>
<th>Resolution</th>
<th>Compset(s) Used</th>
<th>CAM Physics</th>
<th>CAM Dynamics</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAMAtmosphere Data Ocean</td>
<td>CAM3.6.73</td>
<td>1.9x2.5</td>
<td>N/A</td>
<td>CAM3.5.1</td>
<td>FV</td>
</tr>
<tr>
<td></td>
<td>CAM4.1.08</td>
<td></td>
<td></td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td></td>
<td>CAM4.1.13</td>
<td></td>
<td></td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td></td>
<td>CESM 1.0.1</td>
<td>1.9x2.5</td>
<td>F2000</td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td></td>
<td>CESM 1.0.beta22</td>
<td>1.9x2.5_1.9x2.5</td>
<td></td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td>CAMAtmosphere Slab Ocean</td>
<td>CESM 1.0.1</td>
<td>1.9x2.5_gx1v6</td>
<td>E2000</td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td></td>
<td>CESM 1.0.beta22</td>
<td></td>
<td></td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td>Data Atmosphere POP2 Ocean</td>
<td>CESM1.0.beta22</td>
<td>1.9x2.5_gx1v6</td>
<td>G_NORMAL_YEAR</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>CESM1.0.4</td>
<td></td>
<td></td>
<td>CAM4</td>
<td>FV</td>
</tr>
<tr>
<td>CAMAtmosphere POP2 Ocean</td>
<td>CESM 1.0.4</td>
<td>1.9x2.5_gx1v6</td>
<td>B_1850-2000_CN</td>
<td>CAM4</td>
<td>FV</td>
</tr>
</tbody>
</table>


UQ Methodology: UQ Pipeline.

- Use a s/w tool called the LLNL UQ Pipeline (UQP):
  - Runs, organizes, analyzes, & assesses uncertainties in ensemble-of-models calculations.

- User required to supply two Python interface modules:
  - A UQP/Application Data file:
    - Contains settable data items for UQP & Application.
    - Implemented as a set of Python data dictionaries.
  - An Application Interface file, which must provide four functions:
    - prep_ensemble(), prep_run(), post_run(), & post_ensemble().

- An automated capability to build & run the application is also a prerequisite.
UQP: End-to-end, automated, scientific workflow for UQ analysis.

- **Initiate**
  - Define input & initial constraints
    - parameter PDFs / constraints
    - experimental data
    - knowledge of physics problem
    - ...

- **Start**

- **UQP**
  - Sample Generator Stack
  - Aggregation of Adaptive Sample Refinement
  - High Fidelity Surrogate Model?
    - Yes
    - Surrogate Model Constructor Stack
      - Create Training Ensemble Database
      - Statistical Filtering, Sensitivity Analysis, Dimensional Reduction,…
    - No
      - New Experimental Data, Knowledge
      - Analysis Stack
      - Refined predictions with UQ PDFs, parameter PDFs, …
  - Decision Stack
    - Yes
      - Scheduler
        - Schedule Monitor Recover
        - Simulate
        - …
    - No
      - Start

- **Finish**
UQ Methodology: UQP/CESM Interface module details.

- **prep_ensemble()**: 
  - Establishes general CESM framework: 
    - Directory structure, namelist templates, copies files, etc. 
  - Configures CESM, including processor layout. 
  - Builds CESM.

- **prep_run()**: 
  - Sets up a specific ensemble run. 
  - Updates namelist templates with parameter values from UQP.

- **post_run()**: 
  - Computes diagnostics. 
  - Stores/cleans up output files.

- **post_ensemble()**: 
  - Currently just use UQP default analysis.
UQ Methodology:
UQP/CESM Data module example (1).

```
uqdd = {'adjust_code_concurrency': True,
    'bank': 'uqsci',

    'code_concurrency': 4,      # Number of concurrent runs.
    'max_failed_runs': 4,
    'setup_depend_job': True,

    'uqmgr_mode': 'batch',      # [batch|debug|interactive]
    'uqmgr_timeout': 960}       # Job time limit (min.).
```

- Only need to list items that you want to change from their default values.
- Items that a user might change:
  - Brown text: infrequently
  - Red text: fairly often
### UQ Methodology:
**UQP/CESM Data module example (2).**

```python
aprd = {'exe_name': 'ccsm.exe'}

aprd['params_avail'] = {
    'cldfrc_rhminh': [0.65, 0.80, 0.85],
    'cldfrc_rhminl': [0.80, 0.91, 0.99],
    'cldopt_rliqice': [8.4, 14.0, 19.6],
    'cldopt_rliqland': [4.8, 8.0, 11.2],
    'cldopt_rliqocean': [8.4, 14.0, 19.6]
}

aprd['params_to_use'] = {
    ('cldfrc_rhminl', 'lin 0 0 1'),
    ('cldopt_rliqland', 'lin 0 0 1'),
    ('cldopt_rliqocean', 'lin 0 0 1')
}
```

- **Items that a user might change:**
  - Brown text: infrequently
  - Red text: fairly often

---

Library of UQ parameters available; with low, default, & high values.

Actual UQ parameters to use in a study; often a subset of available parameters.
UQ Methodology: CESM build/run script.

- Created a standalone comprehensive Python CESM build/run script:
  - Calls the standard CESM scripts.
  - Frees the user from knowing much about CESM.
  - Completely data driven; flexible.

- Contains additional functionality that the Application Interface module uses when conducting UQ studies.
  - Here the UQP does the actual execution of the runs.
  - All the needed data now comes from the UQ Data module.

- Currently script is ~2K lines of Python code.
UQ Methodology: CESM modifications.

- CESM source code:
  - Mods to 22 files to add needed namelist input capability.

- CESM scripts:
  - General mods to machine-dependent CESM script utilities.
  - `ccsm_buildexe.csh / ccsm_postrun.csh / getTiming.csh`:
    - Commented out all default “gzips” & copying of log files.
    - Changed default handling of timing files.
  - Small mod to `pop2_in_build.csh` to keep pop2 diagnostic files from being overwritten.
UQ Methodology: Putting it all together.

- Determine CESM parameters of interest & their ranges.
  - Add any needed namelist functionality to CESM.

- Implement UQP/CESM Data & Interface modules.
  - Including any post-run diagnostics as desired.
  - Makes use of UQ functionality provided by CESM build/run script.

- For each study:
  - Make any desired changes to UQ Data file settings.
    - Generally not many.
  - Run the study:
    - Pretty much a push-button operation for a study that may contain hundreds of independent runs.
Concluding comments (1).

- Differences in how the various CESM components do their namelist input can be considerable.
- Self-discovery as to how the CESM scripts work “under the covers” is difficult.
- The pop2 ocean model has been somewhat problematic for us:
  - Default output data settings can be quite voluminous:
    - Model crashed when we tried to cut down on this.
  - Appears to be quite sensitive to its parameter settings:
    - Solver fails to converge & floating point exceptions.
  - Initially, we unknowingly over-wrote some diagnostic files.
Concluding comments (2).

- System issues are a major source of problems:
  - Node failures, Lustre disk issues, mpi problems, orphaned system nodes, …
  - Ran into one quite difficult bug:
    - One core on one node was intermittently flipping the least significant bit during a CESM floating point calculation.
    - Ever since, we have done an additional 2 month run with each longer run & check for a bit-for-bit match.