OpenEarthscape

A community platform for computational modeling of earth-surface processes

Gregory E. Tucker,[1,2] Eric W.H. Hutton,[3] Albert Kettner,[3] Julia Moriarty,[3,4] Irina Overeem,[3,2] Mark Piper,[3] Tian Gan,[3] Benjamin Campforts,[3] Scott Peckham,[3] Maria Stoica,[3] Lynn McCready,[3] Nicole Gasparini,[5] Allison Pfeiffer,[6] Erkan Istanbulluoglu,[7] David Gochis,[8] and Ryan Cabell[8]

1 - Cooperative Institute for Research in Environmental Sciences (CIRES), Univ. of Colorado Boulder. 2 - Dept. of Geological Sciences, Univ. of Colorado Boulder. 3 - INSTAAR, Univ. of Colorado Boulder. 4 - Dept. of Atmospheric & Ocean Sciences, Univ. of Colorado Boulder. 5 - Dept. of Earth & Environmental Sciences, Tulane University. 6 - Dept. of Geology, Western Washington Univ. 7 - Dept. of Civil & Environmental Engineering, Univ. of Washington. 8 - National Center for Atmospheric Research.

openEARTHSCAPE

OpenEarthscape is suite of community numerical models, analysis tools, standards, documentation, and cloud resources for research in earth-surface processes.

This poster highlights several elements of OpenEarthscape.

Science themes addressed by OpenEarthscape.

EIT = Earth in Time report (2020)

SC = Sea Change report (2015)

Terrestrial   Both   Coastal/Marine

OpenEarthscape & CSDMS

community surface dynamics modeling system

OpenEarthscape enhances cyberinfrastructure of the **Community Surface Dynamics Modeling System** (CSDMS). CSDMS is an NSF-supported research facility that supports computational modeling in earth-surface science by engaging *community*, providing *computing* resources, and promoting *education*.

https://csdms.colorado.edu

Explore Earth's surface with community software

Read more about CSDMS, its history, goals, and technology in:
Tucker et al. (2022) Geoscientific Model Development,
https://doi.org/10.5194/gmd-15-1413-2022

Landlab Toolkit

https://landlab.github.io

Landlab is an open-source Python package for building and exploring grid-based numerical models. Features include:
o Graph-based **grid** objects
o Modeling with **components**
o Numerical, I/O, & other **utilities**

```
import numpy as np
from landlab import HexModelGrid, imshow_grid
from landlab.components import (FlowAccumulator, LinearDiffuser,
                                StreamPowerEroder)

grid = HexModelGrid((141, 61), spacing=10.0, node_layout='rect')
topo = grid.add_zeros('topographic__elevation', at='node')
np.random.seed()
topo[grid.core_nodes] += np.random.randn(grid.number_of_core_nodes)

fa = FlowAccumulator(grid)
ld = LinearDiffuser(grid, linear_diffusivity=0.01)
sp = StreamPowerEroder(grid, K_sp=0.0002)

baselevel_rate = 0.001
timestep_size = 100.0

for _ in range(400):
    topo[grid.core_nodes] += baselevel_rate * timestep_size
    fa.run_one_step()
    ld.run_one_step(timestep_size)
    sp.run_one_step(timestep_size)

imshow_grid(grid, topo, var_name='Elevation (m)')
```

Example: a simple landscape evolution model in 18 lines of code

Basic Model Interface

https://bmi.readthedocs.io

The Basic Model Interface (BMI) is a language-agnostic API standard for numerical model codes. BMI defines a set of functions for initializing, querying, running, and coupling model codes. The current BMI package includes language specifications and examples for C, C++, Fortran, Python, Java, and (soon) Julia. Through the OpenEarthscape project, the BMI standard is now overseen by a council of active users, with representatives from CSDMS, NOAA, US Geological Survey, Deltares, and eWaterCycle.

`initialize()`

`update()`

`get_value()`

`set_value()`

`finalize()`

Example BMI functions

Taxonomy of modeling tasks (from Tucker et al., 2022)

Babelizer

https://babelizer.readthedocs.io

The babelizer is an open-source Python utility for wrapping model codes that expose a Basic Model Interface (BMI) so that they can be imported as Python packages. The babelizer serves as a language-bridging tool, allowing a researcher to run codes written in C, C++, or Fortran by calling Python BMI functions.

Codes in CSDMS Model Repository, by language

```
[library]
[library.PRMSSurface]
language = "fortran"
library = "bmiprmssurface"
header = ""
entry_point = "bmi_prms_surface"

[build]
undef_macros = []
define_macros = []
libraries = []
library_dirs = []
include_dirs = []
extra_compile_args = []

[package]
name = "pymt_prms_surface"
requirements = ["prms", "prms_surface"]

[info]
github_username = "pymt-lab"
package_author = "Community Surface Dynamics Modeling System"
package_author_email = "csdms@colorado.edu"
package_license = "MIT"
summary = "PRMS6 surface water process component"

[ci]
python_version = ["3.9"]
os = ["linux", "mac", "windows"]
```

Example of a babelizer configuration file (from Hutton et al., 2022, JOSS)

pymt Python Modeling Toolkit

https://pymt.readthedocs.io

The Python Modeling Toolkit (pymt) is an open-source Python package that provides the tools needed for running and coupling models that are equipped with the Basic Model Interface (BMI). Pymt allows Python scripting to function as a framework for coupled modeling and model-data integration.

```
from pymt.models import Cem, Waves

waves = Waves()
cem = Cem()

waves.initialize(*waves.setup())
cem.initialize(*cem.setup())

for time in range(1000):
    waves.update()
    angle = waves.get_value("wave_angle")
    cem.set_value("wave_angle", angle)
    cem.update()
```


Example pymt code that implements a coupled model of wave-field generation and coastline evolution.

EarthscapeHub

OpenEarthscape introduces new CloudBank-hosted Jupyter Hubs for geoscience research & teaching:

lab.openearthscape.org for classes & workshops
jupyter.openearthscape.org for research

Hubs are equipped with CSDMS software as well as teaching and learning materials.

Examples

Landsliding in rugged terrain. *Left: terrain and precipitation are data read into Python using pymt data components. Right: a Landlab code uses the data to model landslide susceptibility.*

Flash flooding. *Hydrograph and water depth modeled using Landlab & a pymt data component.*

Simulating geology from source to sink. *Left: pymt couples onshore erosion model (in C++, with unstructured mesh) with marine delta model (in C, regular grid). Right: Landlab-built simulation integrating rift tectonics, erosion, and sedimentation, from mountain tops to the deep sea.*