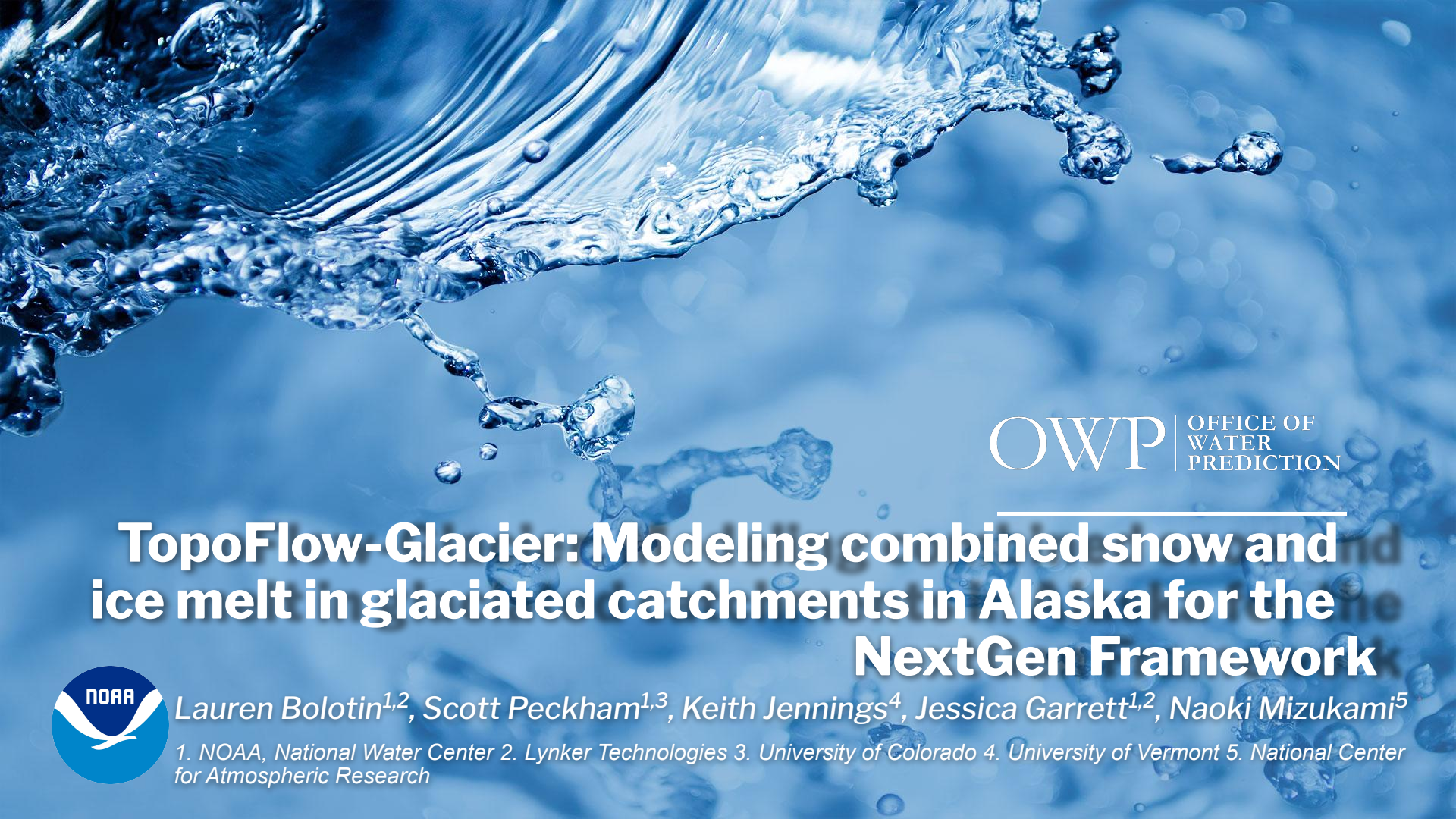


OWP | OFFICE OF WATER PREDICTION

TopoFlow-Glacier: Modeling combined snow and ice melt in glaciated catchments in Alaska for the NextGen Framework

Lauren Bolotin[1,2], Scott Peckham[1,3], Keith Jennings[4], Jessica Garrett[1,2], Naoki Mizukami[5]

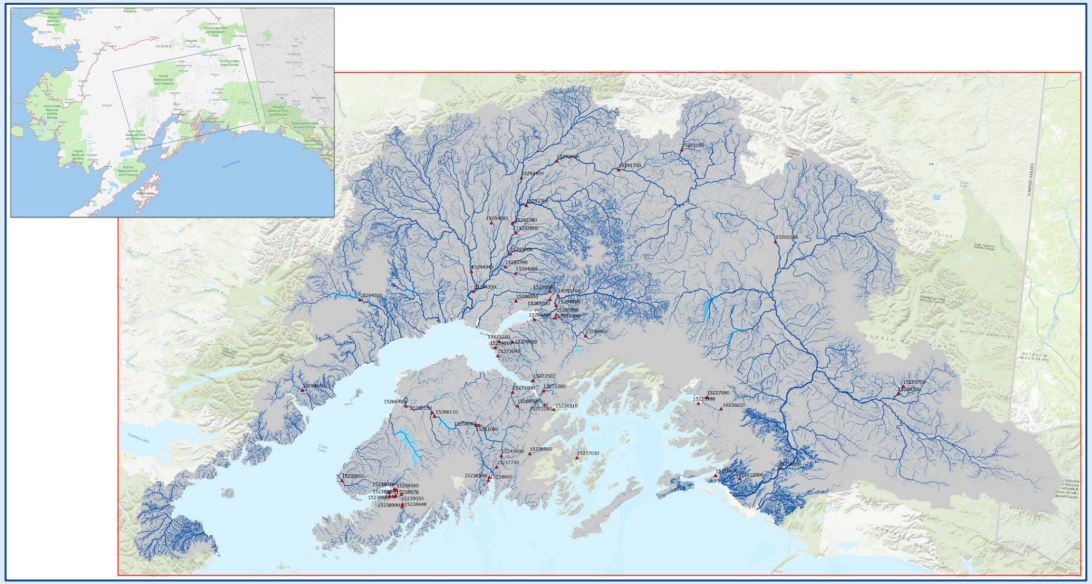
1. NOAA, National Water Center 2. Lynker Technologies 3. University of Colorado 4. University of Vermont 5. National Center for Atmospheric Research

TopoFlow will be used for modeling snow + ice melt in the Alaska domain for NWM v 4.0 (NextGen)

Previously (NWM v 3.0): Crocus snow model in glaciated grid cells

- Written in Fortran
- Complex modeling of several snow processes = more computation time
- Not equipped with BMI

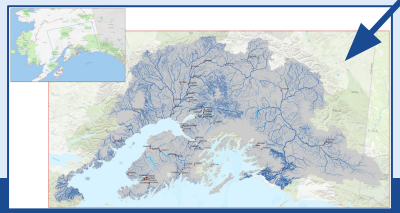
***BMI = Basic Model Interface**
CSDMS: BMI is a standardized set of functions that allows coupling of models to models and models to data.



Cosgrove et al. 2024: Depiction of the NWM Alaska domain released in version 3.0

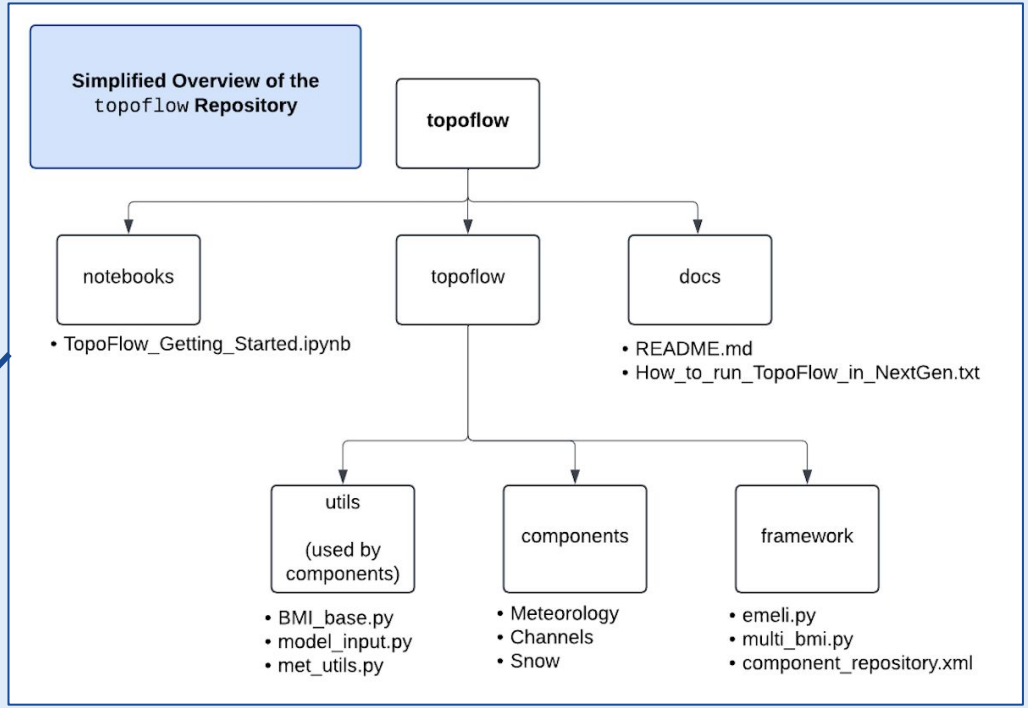
Why TopoFlow for NWM v 4.0?:

- Written in Python
- Existing simple/adaptable snowmelt routines: energy balance, degree day
- Already equipped with BMI
  - Much faster than implementing a non-BMI model



Glaciated grid cells

Cosgrove et al. 2024



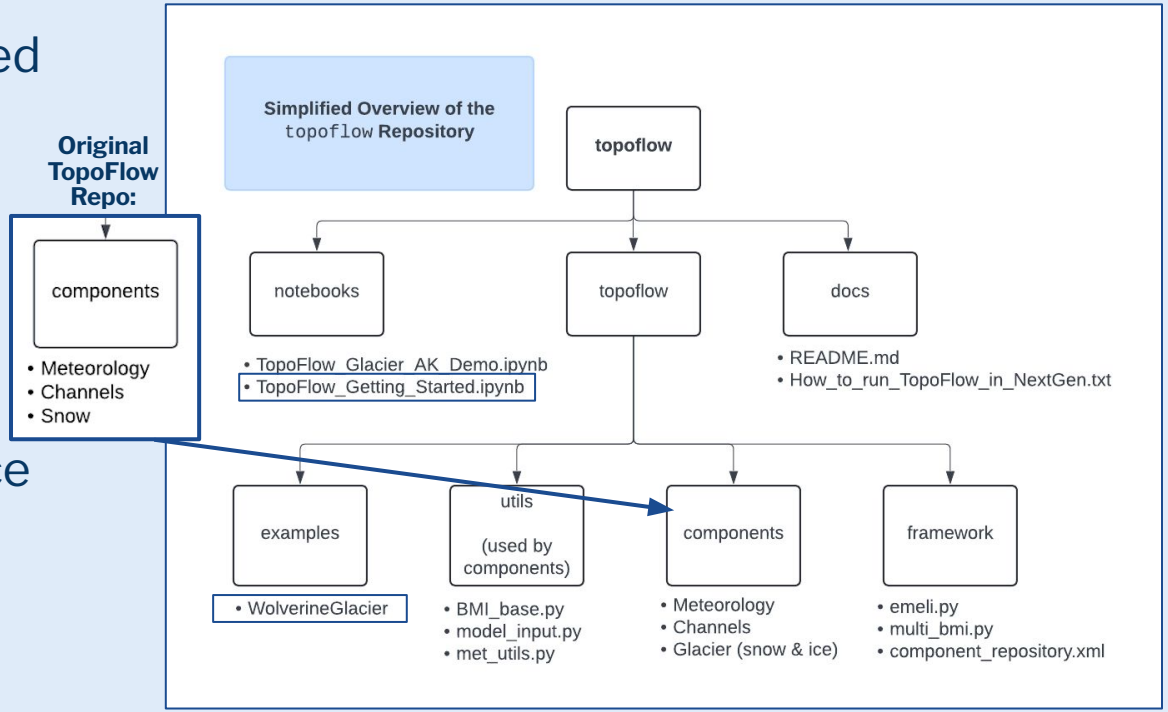
TopoFlow is the 1st NextGen formulation to be forked/developed from a community/public repository

- Multi component, distributed model

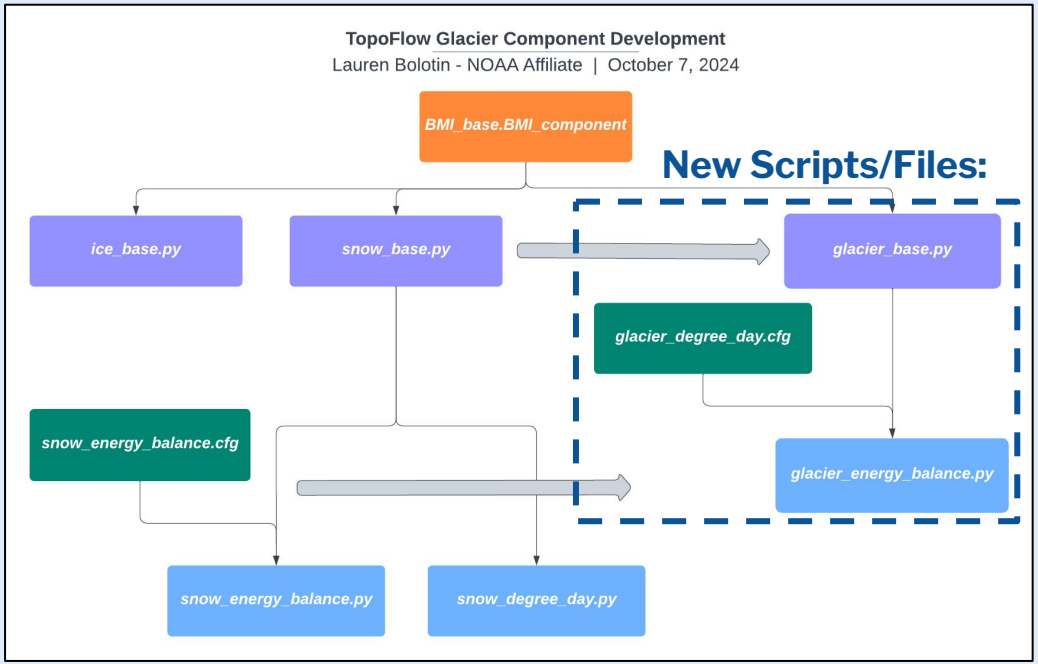
`peckhams/topoflow36`

`NOAA-OWP/topoflow`

- New combined snow **and** ice melt component
- Associated examples



Step 1: Adapt the TopoFlow Snow Energy Balance Routine → Glacier Energy Balance Routine



Snow energy balance routine in TopoFlow

Change parameters to be appropriate for ice

Develop existing snow processes

Glacier energy balance component

Only model ice melt, not accumulation

TopoFlow Model Inputs/Outputs

- **Input: forcing, initial snow/ice depth, initial snow/ice water equivalent,** surface roughness, albedo, etc.
- **Output:** snow/ice meltrate or **combined meltrate**, snow/ice depth/thickness, snow/ice water equivalent, streamflow, etc.

```
#=================================================================
# TopoFlow Config File for: Glacier_Energy_Balance
#=================================================================
# Input
comp_status          | Enabled      | string    | component status (Enabled or Disabled)
n_steps              | 10           | long      | number of time steps
dt                   | 3600.0               | float     | timestep for snowmelt process [sec]
Cp_snow_type         | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
Cp_snow              | 2090.0           | float     | heat capacity of snow [J/kg/K]
Cp_ice_type          | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
Cp_ice               | 2060.0           | float     | heat capacity of snow [J/kg/K]
rho_snow_type        | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
rho_snow             | 300.0          | float     | density of snow [kg/m^3]
rho_ice_type         | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
rho_ice              | 917.0          | float     | density of snow [kg/m^3]
h_active_layer_type     | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
h_active_layer     | 10.0           | float     | thickness of active ice layer [m]
T0_type              | Scalar           | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
T0                   | 0.2              | float     | reference temperature [deg C]
h0_snow_type         | Scalar     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
h0_snow              | 0.0            | float     | depth of snow [m]
h0_ice_type          | Grid     | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
h0_ice               |   __topo/Wolverine1km_glacier_thickness_regrid.rtg        | string     | depth of snow [m]
h0_swe_type          | Scalar       | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
h0_swe               | 0.0            | float     | depth of snow water equivalent (SWE) [m]
h0_iwe_type          | Grid       | string    | allowed input types {Scalar; Grid; Time_Series; Grid_Sequence}
h0_iwe               | __topo/Wolverine1km_glacier_iwe_regrid.rtg           | string     | depth of ice water equivalent (IWE) [m]
```

```
22    h0_ice_type          | Grid      | string     | allowed input types {Scalar;
23    h0_ice               |    __topo/Wolverine1km_glacier_thickness_regrid.rtg
```

Input Data Options:

- Scalar
- Time Series
- Grid
- Grid Sequence

Step 1: Adapt the TopoFlow Snow Energy Balance Routine → Glacier Energy Balance Routine

Original `snow_base.py` **functions:**

```
initialize_cold_content()   # NEE
-----------------------------
update_meltrate()
enforce_max_meltrate()
update_SM_integral()
update_cold_content()
update_swe()
update_density_ratio()
update_depth()
update_total_snowpack_water_volume()
```

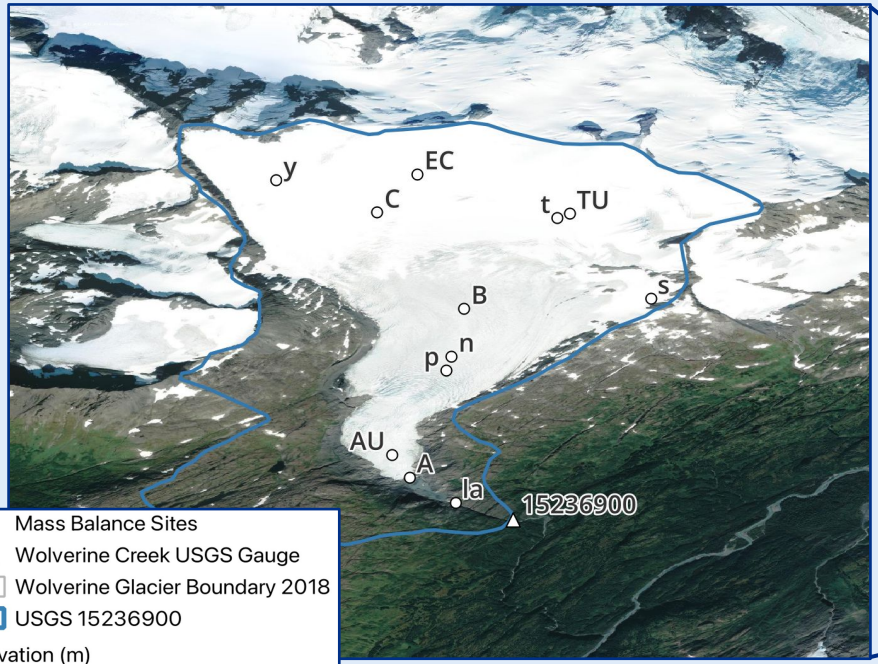
New `glacier_base.py` **functions:**

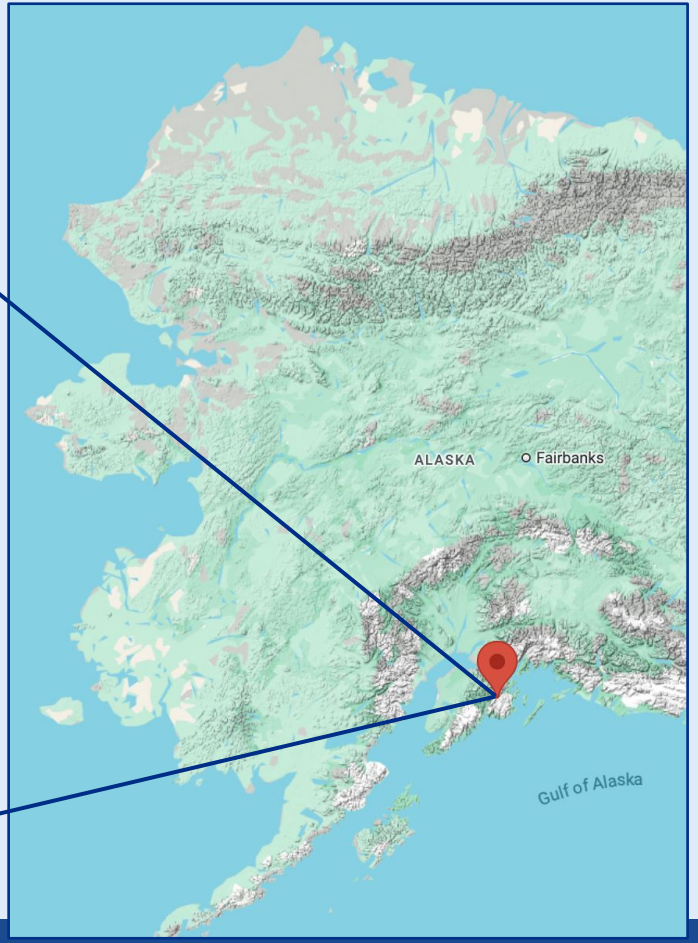
```
initialize_snow_cold_content()
initialize_ice_cold_content()
-----------------------------
update_snow_meltrate()
update_ice_meltrate()
enforce_max_snow_meltrate()
enfore_max_ice_meltrate()
update_SM_integral()
update_IM_integral()
update_snowfall_cold_content()
update_snowpack_cold_content()
extract_previous_swe()
update_swe()
update_density_ratio()
update_swe_integral()
update_iwe()
update_iwe_integral()
extract_previous_snow_depth()
update_snow_depth()
update_ice_depth()
update_total_snowpack_water_volume()
update_total_ice_water_volume()
```

While snow is present in a grid cell, use snow functions and maintain all ice values.

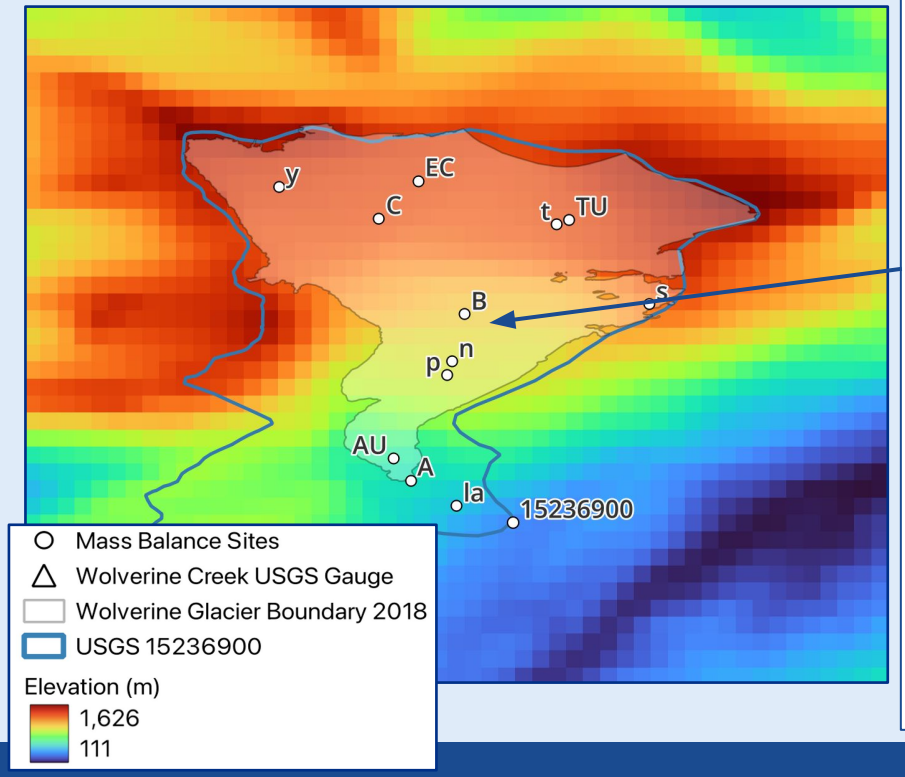
If all snow is melted, use ice functions.

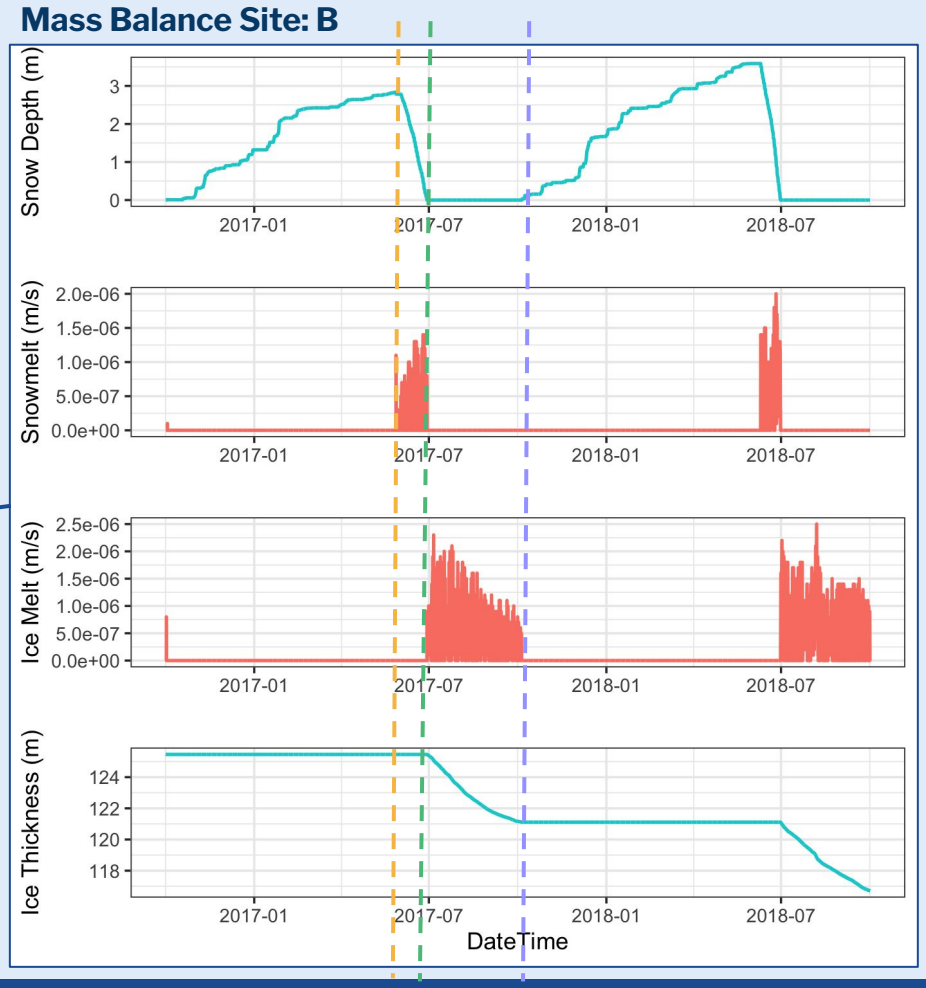
Example TopoFlow Snow/Ice Simulations: Wolverine Glacier





Example TopoFlow Snow/Ice Simulations: Wolverine Glacier





Snowmelt Begins

Snow Accumulation Begins

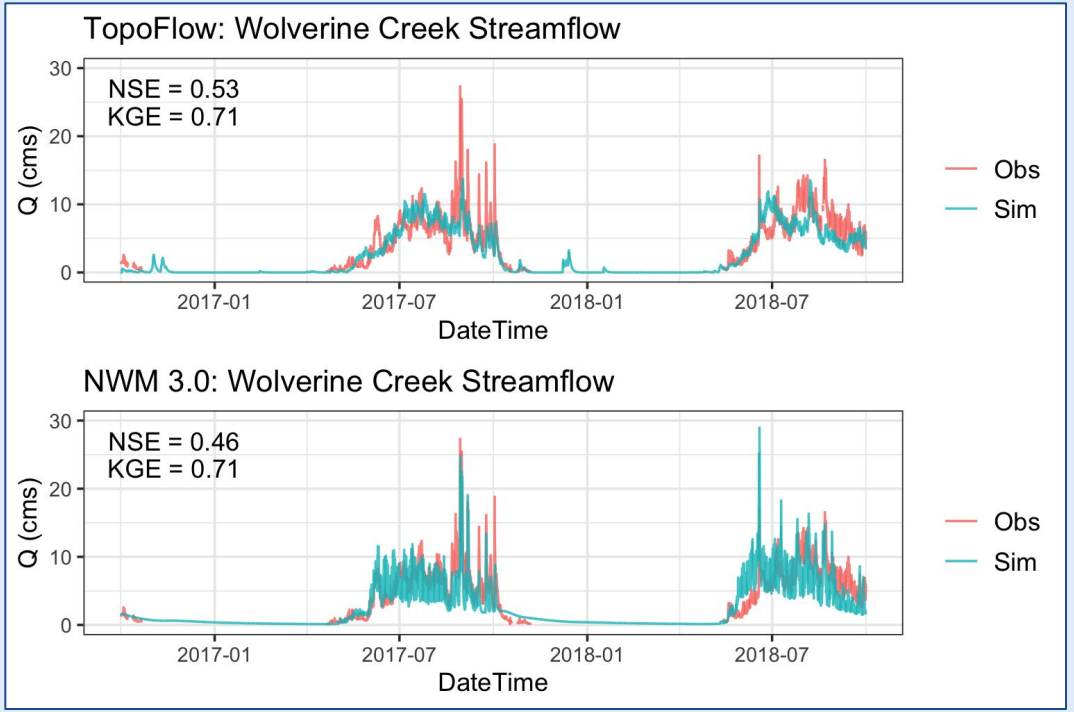
Ice Melt Begins

TopoFlow provides the same KGE and improved NSE compared to NWM 3.0 at Wolverine Creek

***KGE = Kling-Gupta Efficiency**

***NSE = Nash Sutcliffe Efficiency**

- Forcing data comes from Analysis of Record for Calibration (AORC)
- Less noise than NWM 3.0, but dampened high and low flows
- Using TopoFlow channel routing rather than routing capabilities already in NextGen



***Observed hourly data comes from USGS-15236900 (Wolverine Creek, AK)**

Summary

- TopoFlow is the 1st NextGen formulation to be forked/developed from a community/public repository
- Example for future community model contributions to NextGen
- Demonstrates utility of BMI
- First glacier model in NextGen
- In a test basin/glacier, TopoFlow matched or improved simulation performance compared to NWM 3.0

Possible Future Research Directions:

- Large scale testing in NextGen
- Testing coupling with other formulations

Thank You!

Lauren Bolotin (NOAA Affiliate) Lynker

lauren.bolotin@noaa.gov

https://water.noaa.gov

NOAA

OWP | OFFICE OF WATER PREDICTION