Development of the Next Generation UFS Coastal Modeling Framework

Panagiotis Velissariou¹,², Ufuk Turuncoglou⁴, Yunfang Sun¹,³, Saeed Moghimi¹, Ali Abdolali⁵,⁶, Dan Rosen⁴ and Edward Myers¹

¹ NOAA National Ocean Service, Office of Coast Survey (NOAA/OCS)
² University Corporation for Atmospheric Research (UCAR)
³ Earth Resources Technology (ERT/SFI)
⁴ National Center for Atmospheric Research (NCAR)
⁵ U.S. Army Engineer Research and Development Center (ERDC/USACE)
⁶ Earth System Science Interdisciplinary Center (ESSIC/UMD)

Unifying Innovations in Forecasting Capabilities Workshop
Emerging Applications – Coastal & Marine
July 24-28, 2023. Boulder, CO & Online
It takes a village to raise a child ...

NOS Storm Surge Modeling Team
Saeed Moghimi, Panagiotis Velissariou, Soroosh Mani, Yuji Funakoshi, Greg Seroka, Georgios Britzolakis, Zizang Yang, Bahram Khazaei, Yunfang Sun, Fariborz Daneshvar, Edward Myers

Academic partners (>20 PIs, Scientists, Postdocs and PhD students)
- University of Notre Dame
- Virginia Institute of Marine Science
- Argonne National Laboratory
- National Center for Atmospheric Research
- Texas Advanced Computing Center
- Columbia River Inter-Tribal Fish Commission
- Louisiana State University
- Sandia National Laboratories
- University of Massachusetts – Dartmouth
- University of North Carolina at Chapel Hill
- Cooperative Institute for Great Lake Research
- Oregon State University

NOAA and agency partners
- National Ocean Service
  - The U.S. Integrated Ocean Observing System
  - Center for Operational Oceanographic Products and Services
  - National Geodetic Survey
- National Weather Service
  - Office of Science and Technology Integration
  - Environment Modeling Center
  - National Hurricane Center
  - Office of Water Prediction
- Oceanic and Atmospheric Research
  - Great Lakes Environmental Research Laboratory
  - Earth Prediction Innovation Center (EPIC)
- U.S. Geological Survey
- U.S. Environmental Protection Agency
- National Science Foundation

International partners
- Helmholtz-Zentrum Hereon, Germany
- Laboratório Nacional de Engenharia Civil, Portugal
- European Commission Joint Research Centre, Belgium
- International Hydrographic Organization
- United Nations

Industrial and cooperative partners
- NCAR/UCAR
- Spatial Front Inc
- Axiom

Contact: Panagiotis.Velissariou@noaa.gov
Contents

• CoastalApp (Coastal Application)
  ▪ Introduction
    ○ Current Status
  ▪ Workflow
  ▪ Modeling and Data Components

• CoastalApp-testsuite (Tests for CoastalApp)
  ▪ Introduction
    ○ Usage
  ▪ Test Cases

• UFS-Coastal (Coastal Application)
  ▪ Introduction
  ▪ Current Status
  ▪ Issues
  ▪ Future Plans
CoastalApp (Coastal Application)

https://github.com/noaa-ocs-modeling/CoastalApp

Contacts:
- Panagiotis.Velissariou@noaa.gov
- Saeed.Moghimi@noaa.gov

Introduction

CoastalApp is a modeling framework for coastal applications and regional forecasts. It consists of coupled modeling components that link the atmospheric, ocean and terrestrial realms under one common framework. CoastalApp is a flexible and portable modeling system. Flexibility means that additional modeling components can be added with ease and portability means that CoastalApp can be built and run under different computing environments and operating systems.

CoastalApp is based on the ESMF (https://earthsystemmodeling.org/) framework for building a NUOPC/NEMS coupling application that includes two types of components (a) 1-way and 2-way coupled modeling components (model source + NUOPC Cap) and (b) data components (NUOPC Cap only) that pass forcing data, as needed, via NetCDF files to the various models in CoastalApp. The application is based on its predecessor ESMF application ADC-WW3-NUOPC-NEMS (see Moghimi et. al) developed as part of the Coastal Act coupling project to determine wind versus water percentage losses caused by a Named Storm Event.

Accessing the individual modeling components
- ATMESH: https://github.com/noaa-ocs-modeling/ATMESH
- PAHM: https://github.com/noaa-ocs-modeling/PAHM
- ADCIRC: https://adccirc.org/, https://github.com/adccirc/adccirc (requires registration; please send an email request to Crystal Fulcher)
- FVCOM: http://fvcom.smast.umassd.edu/, https://github.com/FVCOM-GitHub
- BARDATA: https://github.com/noaa-ocs-modeling/BARDATA
- WW3DATA: https://github.com/noaa-ocs-modeling/WW3DATA
CoastalApp: System Workflow

By Design a Flexible & Portable System

Preprocessing
Control/Configuration Files Boundary & Initial Conditions

Wave Model

Sea Ice

Baroclinic Effects

Coastal Ocean Model(s)

Inland Hydrology Model

Numerical Weather Model(s)

PaHM

- Navigation support
- Disaster mitigation
- Water Quality
- Sediment Transport

Status
- Implemented
- In development/testing or future capability

Product End User

- NUOPC/ESMF Coupled Models
- 1-way/2-way Model Coupling
- Development in Collaboration Between Federal and Non-Federal Partners

Contact: Panagiotis.Velissariou@noaa.gov

https://github.com/noaa-ocs-modeling/CoastalApp
## CoastalApp: Modeling and Data Components

[https://github.com/noaa-ocs-modeling/CoastalApp](https://github.com/noaa-ocs-modeling/CoastalApp)

The components highlighted in red are not implemented or fully functional

<table>
<thead>
<tr>
<th>Atmosphere</th>
<th>Ocean</th>
<th>Wave</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMESH¹ (implemented)</td>
<td>ADCIRC² (implemented)</td>
<td>WW3DATA¹ (implemented)</td>
</tr>
<tr>
<td>PaHM¹ (implemented)</td>
<td>SCHISM⁴,⁵ (implemented)</td>
<td>WW3³ (implemented)</td>
</tr>
<tr>
<td>Atm. Model (in development)</td>
<td>FVCOM⁶ (implemented)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BARDATA¹ (implemented)</td>
<td></td>
</tr>
</tbody>
</table>

|  | CICE⁷ (in development) |  |
|  | NWM⁸ (in development) |  |

---

1. NOAA/CSDL/CMMB
2. U. of Notre Dame
3. NOAA/NCEP/EMC
4. Virginia Institute of Marine Science
5. Helmholtz-Zentrum Hereon
6. University of Massachusetts – Dartmouth
7. Cooperative Institute for Great Lakes Research
8. NOAA/NWS National Water Center
CoastalApp: Directory Tree

```
git clone --recurse-submodules https://github.com/noaa-ocs-modeling/CoastalApp.git
```

**Support Matrix**

- **Compute Platforms:** HPC clusters, personal desktops and cloud compute resources
- **Operating Systems:** Unix/Linux, MacOSX
- **Build System:** Bash, Module environment(s), Capabilities for building OS missing libraries

**Contacts:**
- Panagiotis.Velissariou@noaa.gov
- Saeed.Moghimi@noaa.gov
CoastalApp-testsuite (Tests for CoastalApp)

Contact: Panagiotis.Velissariou@noaa.gov

NOAA/NOS’ Office of Coast Survey

https://github.com/noaa-ocs-modeling/CoastalApp-testsuite

Introduction

CoastalApp-testsuite contains comprehensive tests for the different modeling components implemented in CoastalApp. The test suite is used to run automated tests for the model and data components after an update in CoastalApp. There are two sets of tests: (a) small scale tests that require very limited compute resources (e.g., the Shinnecock inlet cases) and (b) large scale tests that require extensive compute resources that can be run on a Cluster/HPC environment (e.g., the HSOFS cases). In any case, to run any of these tests the user is responsible to download and compile CoastalApp first.

Job Submission Managers

- **SLURM** with user supplied options
- **PBS** with user supplied options
- **mpirun/mpiexec** with user supplied options

Regression tests for both CoastalApp and UFS-Coastal

Run Sequence

1. Change directory into CoastalApp-testsuite
2. Edit the file regtest_list.dat and uncomment the test cases you want to run
3. Edit (or create) an "environment file" (a sample can be found in templates/env_tests) that contains values for the different options used by the run script. The location of this file by setting the environment variable TESTS_ENV_FILE to point to the location of the newly created file (if env_tests is in the same location as run_all.sh there is no need to set the TESTS_ENV_FILE variable). If most of the option values remain the same between run sequences, it is convenient to have this file in place and only supply a few options to the script

   **OPTIONAL STEP**

4. Run the run_all.sh script to initiate the run sequence for the requested tests
Contact: Panagiotis.Velissariou@noaa.gov

NOAA/NOS' Office of Coast Survey

Multiple “testbed” platforms are supported:
• NOAA RDHPCS (hera)
• TACC (Frontera, Stampede)
• Mississippi State University (Orion, Hercules)
• Cloud (Parallel Works)

git clone https://github.com/noaa-ocs-modeling/CoastalApp-testsuite.git

Component name abbreviations used in the testsuite:
- atm: ATMESH data component
- pam: PAHM model component
- adc: ADCIRC model component
- sch: SCHISM model component
- fvc: FVCOM model component
- ww3: WaveWatch III model component
- ww3data: WW3DATA data component
At this point, only one-way configurations from CoastalApp-testsuite are ported to UFS-Coastal. The two-way configurations will be ported individually from CoastalApp-testsuite to UFS RT framework. The new application is using CDEPS as data component and replaces the ATMESH data component.

ADCIRC and FVCOM model components are under development. ROMS and PAHM are future model component capabilities.
UFS-Coastal: Directory Tree

- ADCIRC-interface
- AQM @ 37cbb7d
- CDEPS-interface
- CICE-interface
- CMEPS-interface
- CMakeModules @ cabd775
- FV3 @ 67e146d
- FVCOM-interface
- GOCART @ b94145f
- HYCOM-interface
- MOM6-interface
- NOAHMP-interface
- SCHISM-interface
- WW3 @ c4b1168
- cmake
- doc/UsersGuide
- driver
- modulefiles
- stochastic_physics @ 3bfa446

Contacts:
- Ufuk Turuconglu (turuncu@ucar.edu)
- Panagiotis.Velissariou@noaa.gov
- Saeed.Moghimi@noaa.gov

Active Model Interfaces
- ADCIRC
- FVCOM
- SCHISM

Active Regression Tests
- ADCIRC (testing)
- FVCOM (testing)
- SCHISM (fully functional)

https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app
UFS-Coastal: Current Status

https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app

- **UFS-Coastal** is now a fork of UFS Weather Model (branch: feature/coastal_app)
  - It is maintained under [Ocean Modeling Collaboration](https://github.com/oceanmodeling) GitHub organization to have better collaboration with community

- Currently, following components are ported from CoastalApp to UFS-Coastal:
  - **ADCIRC**: unstructured mesh model to solve free surface circulation and transport
  - **FVCOM**: unstructured mesh finite volume ocean model
  - **SCHISM**: unstructured mesh model to solve baroclinic circulation across creek-lake-river-estuary-shelf-ocean scales
  - ATMESH custom data component is replaced with [CDEPS](https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app)
UFS-Coastal: Current Status (cont’d)

https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app

- UFS Weather Model uses CMake to build components
  - To that end, all models are wrapped with CMake build interface under UFS-Coastal
- SCHISM model NUOPC “cap” is slightly modified to work with CMEPS mediator. All the components under CoastalApp was coupled through the NUOPC connectors
- Tests under CoastalApp-testsuite have been ported to RT system one-by-one.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># ADCIRC tests</td>
<td>COMPILE</td>
<td>1</td>
<td>intel</td>
<td>-DAPP=CSTLA -DADCIRC_CONFIG=PADCIRC -DCOUPLED=ON</td>
</tr>
<tr>
<td># FVCOM tests</td>
<td>COMPILE</td>
<td>2</td>
<td>intel</td>
<td>-DAPP=CSTLW</td>
</tr>
<tr>
<td># SCHISM tests</td>
<td>COMPILE</td>
<td>3</td>
<td>intel</td>
<td>-DAPP=CSTLS -DNO_PARMETIS=OFF -DOLDIO=ON</td>
</tr>
<tr>
<td></td>
<td>RUN</td>
<td>coastal_ike_shinnecock_atm2sch</td>
<td></td>
<td>baseline</td>
</tr>
<tr>
<td># WW3 tests</td>
<td>COMPILE</td>
<td>4</td>
<td>intel</td>
<td>-DAPP=CSTLW</td>
</tr>
</tbody>
</table>
UFS-Coastal: Issues

https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app

- We are actively working on ADCIRC and FVCOM model components to make their NUOPC “caps” compatible with CMEPS mediator
  - CMEPS uses mesh as a representation of model grid/mesh
  - CMEPS assumes that all the import and export fields connected from the component on element location (the default of ESMF_FieldCreate() is node)
  - ESMF has no support for ghost elements at this point

- We need to remove ghost elements from ADCIRC and FVCOM meshes and define import and export fields on element locations

- Need to generalize forcing provided by the CDEPS (ATMESH data atmosphere mode)
  - All the test cases that are aimed to simulate Hurricane Florence or Ike need to be forced with same forcing. This will allow us to compare the results and find possible issues with the configurations
UFS-Coastal: Plans

https://github.com/oceanmodeling/ufs-coastal/tree/feature/coastal_app

- Finalize work related to ADCIRC and FVCOM coupling interface and make them compatible with CMEPS
- Update already ported regression tests and create baseline for them
  - These tests will run regularly against the development in UFS Weather Model and in case of syncing UFS-Coastal with authoritative repository
- Add new tests for UFS-Coastal model components to cover the model components at least with single coupled test
- Modernize UFS-Coastal build system by using ESMX generic driver
- Implement isolated CI testing to each model components
  - This will benefit from nuopc-comp-testing composite GitHub action
  - Extend/improve nuopc-comp-testing composite GitHub action if it is required
Thank you for your attention!
CoastalApp: System Layout and Usage

Usage: “build.sh” [-|--option1|--|option_value1] [-|--option2|--|option_value2] ...

- h: help -h --help
  Show this help screen.

-c|--clean|--clean |--space|"0|1|yes|no" (OPTIONAL).
  Only clean the already compiled CMake build system.
  Defaults: 0|no.

-compiler|--compiler |--space|"compiling_system" (OPTIONAL).
  The compiling system to use (gnu, intel, pgI).
  Defaults: intel.

-configuration|--configuration |--space|"component_list" (OPTIONAL).
  The component(s) to use (ATMESH, PAHm, ADcIRC, NEMSy, WWData, WW3, BARData).
  Defaults: “ADcIRC WW3DATA ATMESH”.

-build_exec|--build_exec |--space|"executable_list" (OPTIONAL).
  The executables to build (e.g., “build_exec=podc build pahm”).
  Defaults: none.

-M|--M|--|space|”M” (OPTIONAL).
  Define the number of make jobs to run simultaneously.
  Defaults: 1.

-par|--par|--parallel|--parallel |--space|"0|1|yes|no" (OPTIONAL).
  Activate the use of parallel compilers.
  Defaults: 0|yes.

-os|--os|--space|"OS string" (OPTIONAL).
  The name of the operating system.
  Supported OSes: linux, macosx.
  Defaults: current OS.

-platform|--platform|--platform|--|space|"platform" (OPTIONAL).
  The name of the compute HPC platform to consider.
  Selecting a platform, environment modules specific to that platform are loaded
  and corresponding environment variables are set.
  Supported platforms: custom, linux, macosx, cheyenne, gaea, hera, jet, orion, stampede, wcoss,
  mistral, strain.
  Defaults: OS.

-v|--verbose|--verbose|--|space|"a,b,w,i,j,m,n" (any combination, OPTIONAL).
  Enable verbosity in the make files during compilation.
  a (all) : all types of debugging output are enabled
  n (none) : disable all debugging currently enabled
  b (basic) : basic debugging and whether the build was successfui or not
  v (verbose) : a level above basic
  i (implicit) : prints messages describing the implicit rule sequences for each target
  j (jobs) : prints messages giving details of the invocation of specific sub-commands
  m (makefile) : enables messages while rebuilding makefiles
  Default: none.
CoastalApp: ModuleFiles

Typical modulefile

Compiler one of [gnu, intel, pgi]

Load Modules

Set Environment

Finalize (if needed)
CoastalApp: Build Workflow

Usage: build.sh [[-|--]option1=[space][option_value1]] [[-|--]option2=[space][option_value2]] ...

-h|--help|--h|--help
Show this help screen.

-compiler|--compiler [=|space] "compiling_system" (OPTIONAL).
The compiling system to use (gnu, intel, pg1).
Default: intel.

-component|--component [=|space] "component_list" (OPTIONAL).
The component(s) to use (ATMESH WRF HWRF PAHM ADCIRC SCHISM FVCOM ROMS NWM WW3DATA WW3 ).
Default: "ADCIRC WW3DATA ATMESH".

-par|--par|--parallel|--parallel [=|space] "0|1|yes|no" (OPTIONAL).
Activate the use of parallel compilers.
Default: 1|yes.

-plat|--plat|--platform|--platform [=|space] "platform" (OPTIONAL).
The name of the compute HPC platform to consider.
Selecting a platform, environment modules specific to that platform are loaded
and corresponding environment variables are set.
Supported platforms: custom, linux, macosx, cheyenne, gaea, hera, jet, orion, stampede, wcoss,
mistral, strand.
Default: OS.
As a part of the project, ESMF/NUOPC layer is extended to include Generic NUOPC Driver layer, which is called as Earth System Model eXecutable (ESMX).

Aims to accelerate development of new NUOPC-based systems.

The ESMF 8.5.0 (targeted for end of July 2023) will have improved version of ESMX driver.

- It uses YAML based configuration files for build (ESMX_Builder) and run.

The UFS-Coastal build system will be restructured to replace existing UFS NUOPC driver with new generic driver implementation.

---

**Example: DATM+LND**

```plaintext
application:
  disable_comps: ESMX_Data
  link_paths: /home/runner/.spack-ci/view/lib
  link_libraries: piof

components:
  datm:
    build_type: none
    install_prefix: /test/app/cdeps/install
    libraries: datm dshr streams cdeps_share
    fort_module: cdeps_datm_comp.mod
  noahmp:
    build_type: none
    install_prefix: /test/app/noahmp/install
    fort_module: lnd_comp_nuopc.mod
```

---
UFS-Coastal: Testing NUOPC Components

- New hierarchical testing capability for NUOPC components: nuopc-comp-testing

https://github.com/esmf-org/nuopc-comp-testing

DATM+LND example

VM / Containerized Environment

Install core development tools (apt-get install)

Install dependencies * (Spack)

Install CDEPS data components (i.e. DATM)

Install model component that needs to be tested (i.e. Noah-MP)

Create model executable (generic driver NUOPC/ESMX)

Create run directory for test

Get inputs (wget, ftp, Amazon S3 bucket) via Python *

Create namelist and configuration files (ParamGen)

Execute tested configuration

Baseline Check (PASS/FAIL) *

* Steps that uses GitHub cache mechanisms