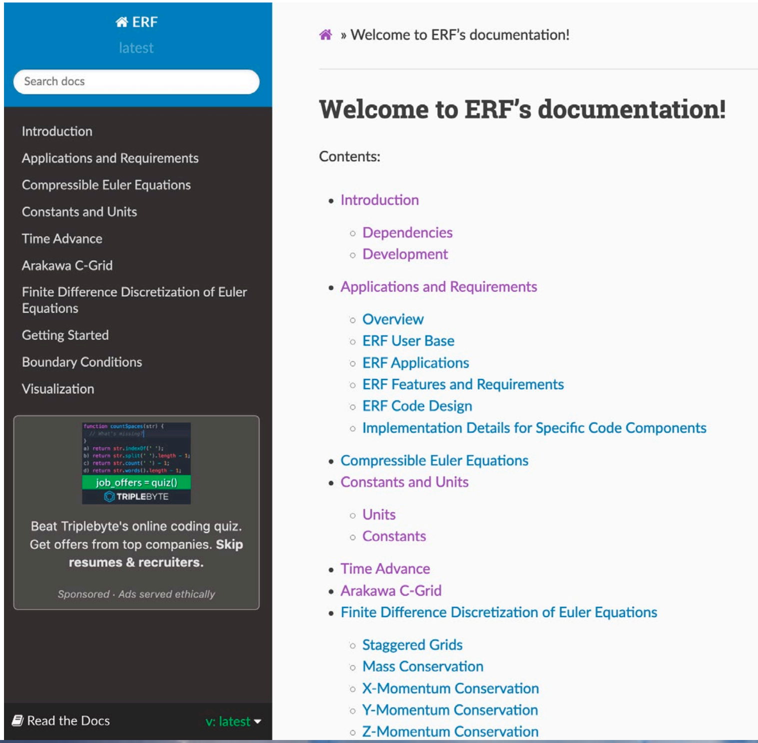
U.S. DEPARTMENT OF ENERGY

Office of ENERGY EFFICIENCY & RENEWABLE ENERGY

2021 PROJECT PEER REVIEW
U.S. DEPARTMENT OF ENERGY
WIND ENERGY TECHNOLOGIES OFFICE

T17 – Energy Research and Forecast modeling (ERF)



Tech R&D – Atmosphere to electrons
Jeffrey D. Mirocha
LLNL

4 August 2021

FY21 Peer Review – ERF Project Overview

Project Summary:

- The ERF project will facilitate multiscale atmospheric-wind plant simulation within a wide range of wind energy applications by developing a modern atmospheric downscaling code to accurately and efficiently exchange flow information across scales between weather (energy) and the wind plant on next-generation high-performance computing architectures.
- Collaboration between LLNL, ANL, NREL, PNNL and NCAR

Project Start Year: [FY19 (Q4)]
Expected Completion Year: [FY 24]
Total expected duration: [5] years

FY19 - FY20 Budget: 1.35M

Key Project Personnel: Jeff Mirocha (LLNL), Rao Kotamarthi (ANL), Eliot Quon (NREL), Bill Gustafson (PNNL), Branko Kosovic (NCAR)

Key DOE Personnel: Shannon Davis

Project Objective(s): 2019-2020

- Rescope a tractable project plan based on results of project workshop, in light of changes to emerging computational architectures
- Develop detailed Applications, Requirements and Design documents to guide code development toward the highest wind energy impacts
- Leverage the AMReX mesh refinement framework to maximize efficient use of multiple computational architectures
- Set up open Github for code development tracking, documentation, and sharing of the code with the community
- Develop appropriate verification and validation tests, and automate

Overall Project Objectives (life of project):

- A modern code base to seamlessly couple mesoscale energy flows with microscale wind plant simulation to advance wind energy deployment.

ERF Project Impact

- Multiscale environmental interactions influence the entire life cycle of wind power generation, resource characterization → design → operation → integration.
- Broader incorporation of multiscale atmosphere/wind plant simulation into wind energy workflows has potential for far reaching, industrywide impacts.
- ERF enables efficient multiscale atmospheric simulation on multiple emerging HPC architectures (via the AMReX adaptive mesh framework).
  - Emerging high-performance computing (HPC) hardware is shifting to graphics processing units (GPUs).
  - Models in use or under development either are not multiscale, or not GPU – compatible.
- ERF will seamlessly couple with the AMRWind microscale wind plant code (also built upon AMReX) to enable high-fidelity wind plant simulations in diverse atmospheric flow regimes.
- ERF will assimilate procedures from related WETO projects (e.g. Mesoscale-Microscale Coupling, Offshore Wind Resource Science, ...).
- Open-source code with extensive documentation and test cases to facilitate widespread adoption.
- ERF has the potential to serve as a foundational code across a broad swath of WETO projects, as well as multiple research and operational centers.
- Potential for wide applicability across weather-dependent renewable energy sectors (grid integration, energy storage, hybrid plant operation, other services).

Program Performance – Scope, Schedule, Execution

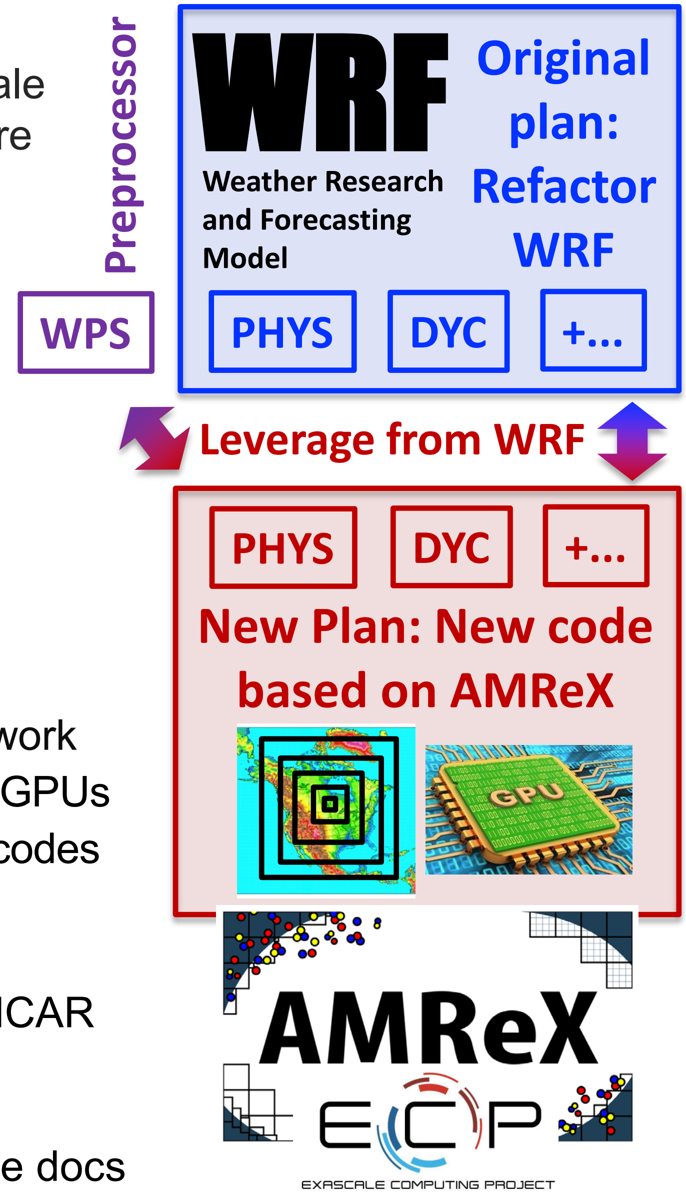
Obstacle
- Original Plan (2018): Refactor WRF for more efficient multiscale computation targeting Intel's Many Integrated Core architecture
- (2018-2019) HPC landscape rapidly shifted toward GPUs
- Project kickoff workshop (FY20 Q1): Consensus –WRF's software is not compatible with GPU architectures

Opportunity
- Design ERF from ground up (but not starting from scratch)
  - Develop a modern, efficient and flexible software (in C++)
  - Target wind energy research needs directly

Challenge
- Re-envision/rescope project within budget constraints (FY20)

Solution
- Build ERF upon the AMReX sdaptive mesh refinement framework
  - Built-in abstractions for multiple HPC architectures, including GPUs
  - C++ and Fortran ➔ rapid buildout using modules from other codes
  - Exascale Computing Project support ensures maintainability

Project Execution
- Multi-lab coordination among LLNL, ANL, PNNL, NREL and NCAR
  - Weekly team technical meetings
  - Monthly team meetings with management
  - Code and documentation shared through Github and Google docs



Project Performance – FY21 Activities: ERF Github, documentation

ERF development: https://github.com/erf-model

- ERF model source code
- Verification and validation test cases
- Extensive documentation (readthedocs)

ERF

latest

Search docs

» Welcome to ERF's documentation!

Welcome to ERF's documentation!

Contents:

Compressible Euler Equations

ERF advances the following set of equations:

$$\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{u}),$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} = -\nabla\cdot(\rho\mathbf{u}\mathbf{u}+pI)+\rho\mathbf{g},$$

$$\frac{\partial(\rho\theta)}{\partial t} = -\nabla\cdot(\rho\mathbf{u}\theta),$$

$$\frac{\partial(\rho A)}{\partial t} = -\nabla\cdot(\rho\mathbf{u}A),$$

The relationship between potential temperature and temperature is given by

$$\theta = T(\frac{p_0}{p})^{R_d/c_p}$$

and we use the following equation of state:

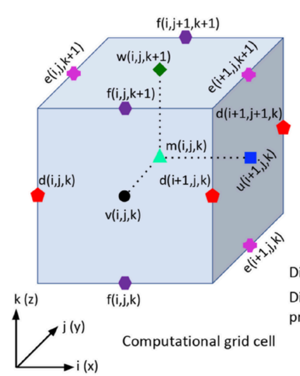
$$p = \rho R_d T;$$

which can also be written in terms of θ as

$$p = (\rho R_d \theta / p_0^{R_d/c_p})^{\gamma}$$

Here ρ, T, θ, and p are the density, temperature, potential temperature and pressure, respectively; these variables are all defined at cell centers. A is an advected quantity, i.e., a tracer, also defined at cell centers $\mathbf{u}$ and $(\rho\mathbf{u})$ are the velocity and momentum, respectively, and are defined on faces. The gravitational vector is denoted by $\mathbf{g}$.

Arakawa "C" grid prognostic and diagnostic variable distribution



Computational grid cell

- u = x-direction velocity, scalar flux s_1, staggered in i
- v = y-direction velocity, scalar flux s_2, staggered in j
- w = z-direction velocity, scalar flux s_3, eta, staggered in k
- m = temperature, pressure, density, vorticity ω_{ij}, i=j
- d = vorticity ω_{12}, staggered in i and j
- e = vorticity ω_{13}, staggered in i and k
- f = vorticity ω_{23}, staggered in j and k

Vorticity points include deformations and momentum stresses

eta is the vertical coordinate if a terrain-following grid is specified

Divergence of velocity projects to m points at cell centers

Divergence of fluxes influencing tendencies of a given prognostic vatiable project to that variable's nodes

The following page shows the above variables in several cut planes with indexing.

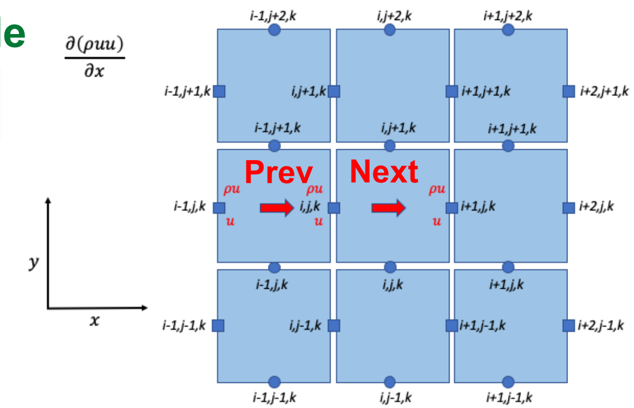
Project Performance – FY21 Activities: ERF code development

Correspondence between equations, notation, and source code

- Governing equation (u-momentum): $\frac{\partial(\rho\mathbf{u})}{\partial t} = -\nabla\cdot(\rho\mathbf{u}\mathbf{u} + pI) + \rho\mathbf{g},$
- Discretized equation:

$$
\begin{aligned}
(\rho u)^{n+1}_{i,j,k} = (\rho u)^{n}_{i,j,k} - \Delta t \Big\{ & \frac{1}{2\Delta x}\Big[\big((\rho u)^n_{i+1,j,k} + (\rho u)^n_{i,j,k}\big)u^n_{i+\frac{1}{2},j,k} - \big((\rho u)^n_{i,j,k} + (\rho u)^n_{i-1,j,k}\big)u^n_{i-\frac{1}{2},j,k}\Big] \\
& + \frac{1}{2\Delta y}\Big[\big((\rho v)^n_{i,j+1,k} + (\rho v)^n_{i-1,j+1,k}\big)u^n_{i,j+\frac{1}{2},k} - \big((\rho v)^n_{i,j,k} + (\rho v)^n_{i-1,j,k}\big)u^n_{i,j-\frac{1}{2},k}\Big] \\
& + \frac{1}{2\Delta z}\Big[\big((\rho w)^n_{i,j,k+1} + (\rho w)^n_{i-1,j,k+1}\big)u^n_{i,j,k+\frac{1}{2}} - \big((\rho w)^n_{i,j,k} + (\rho w)^n_{i-1,j,k}\big)u^n_{i,j,k-\frac{1}{2}}\Big]\Big\} \\
& - \frac{\Delta t}{\Delta x}\big[p^n_{i,j,k} - p^n_{i-1,j,k}\big]
\end{aligned}
$$

- Source code:
  - Compute fluxes:



- Discretization graphics:

```
// Add advective terms
Real centFluxXXNext, centFluxXXPrev, edgeFluxXYNext, edgeFluxXYPrev, edgeFluxXZNext, edgeFluxXZPrev;
centFluxXXNext = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::next,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_u, solverChoice.spatial_order);
centFluxXXPrev = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::prev,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_u, solverChoice.spatial_order);
edgeFluxXYNext = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::next,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_v, solverChoice.spatial_order);
edgeFluxXYPrev = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::prev,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_v, solverChoice.spatial_order);
edgeFluxXZNext = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::next,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_w, solverChoice.spatial_order);
edgeFluxXZPrev = ComputeAdvectedQuantityForMom(i, j, k, rho_u, rho_v, rho_w, u, v, w, nextOrPrev: NextOrPrev::prev,
                 advectedQuantity: AdvectedQuantity::u, advectingQuantity: AdvectingQuantity::rho_w, solverChoice.spatial_order);
```

  - Time advance:

```
rho_u_upd(i, j, k) += (-dt) * (
                 (centFluxXXNext - centFluxXXPrev) / dx[0] // Contribution to x-mom eqn from flux in x-dir
                +(edgeFluxXYNext - edgeFluxXYPrev) / dx[1] // Contribution to x-mom eqn from flux in y-dir
                +(edgeFluxXZNext - edgeFluxXZPrev) / dx[2] // Contribution to x-mom eqn from flux in z-dir
```

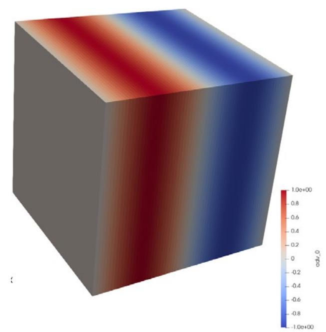
Project Performance – FY21 Activities: ERF testing

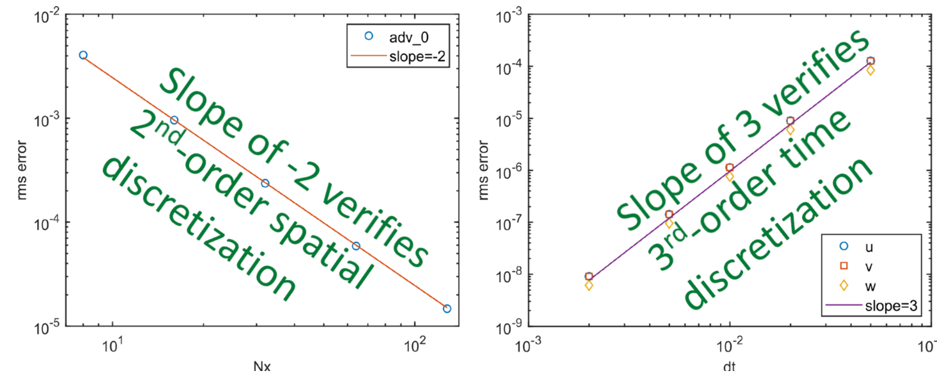
Verification and validation of ERF compressible atmospheric solver code is underway

1. Spatial discretization:

Scalar Advection and Diffusion

- $\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial x} = D\frac{\partial^2 f}{\partial x^2}$
- $f(x,t) = e^{-Dt}\sin(x - at)$





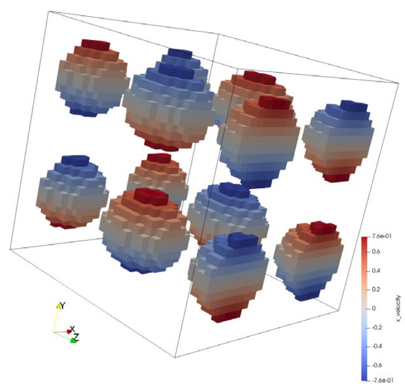
2. Time integration:

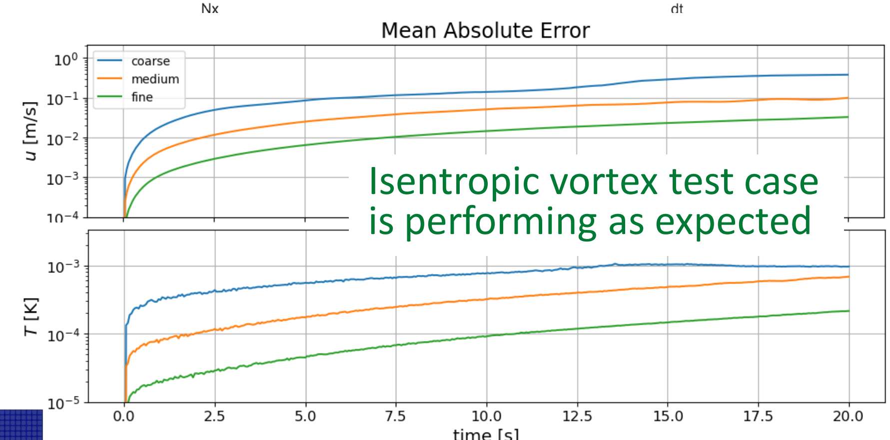
Taylor-Green Vortex

$$u = A\cos ax \sin by \sin cz,$$

$$v = B\sin ax \cos by \sin cz,$$

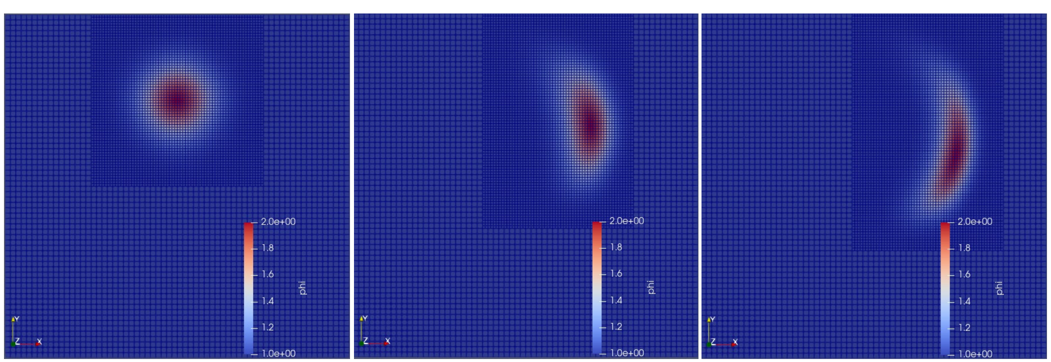
$$w = C\sin ax \sin by \cos cz.$$





3. Adaptive mesh refinement.

Scalar Advection and Diffusion



- ERF code is passing all verification tests
- New tests being developed for each new functionality added
- Test suite is being archived
- Automated testing is being established

Project Performance – Remaining Activities through FY24:

FY21:

Go/No-Go Criterion: Demonstrate ERF in an atmospheric boundary layer (ABL) simulation, either coupled with AMRWind, or with a concrete near-term plan to complete the coupling, **by 7/31/21**.

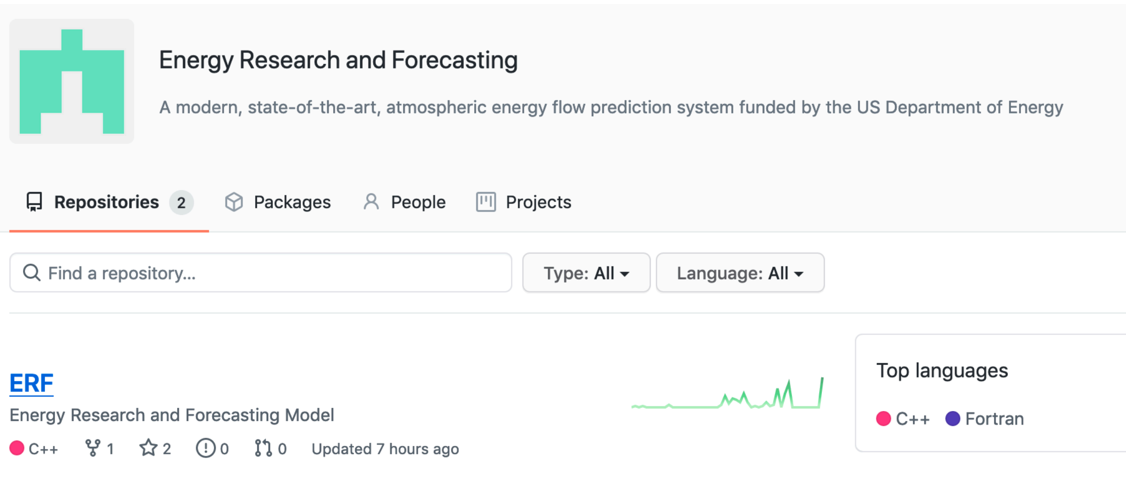
- (Q3) Add Log-Law boundary condition for the surface.
- (Q3) Add pressure gradient/geostrophic forcing for ABL simulation.
- (Q3) Finish addition of Smagorisky subgrid model for large-eddy simulation.
- (Q3) Couple ERF outflow with AMRWind and test initial coupling of the two codes.
- (Q4) Test ERF ABL simulations against expected behaviour, simulation results from WRF.
- (Q4) Assess computational performance improvements using ERF versus WRF.

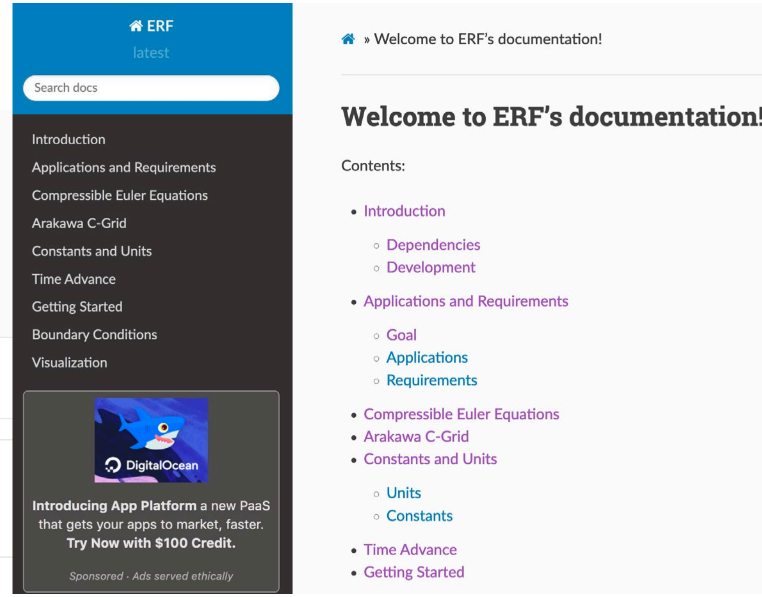
FY22-24:

- (FY22) Incorporate mesoscale physics modules to capture mesoscale energy features.
- (FY22) Incorporate interface to WRF preprocessor to ingest large-scale forcing data.
- (FY23) Add terrain-following coordinates and/or immersed boundaries to represent surfaces.
- (FY23) Incorporate appropriate wave-atmosphere interaction models for offshore settings.
- (FY23-24) Assess performance of ERF versus WRF in multiscale, whole model setups.
- (FY23-24) Assess performance of coupled ERF AMRWind simulations.
- (FY24) A validated source code with extensive documentation, user/developer resources, and relevant test cases to demonstrate capability and facilitate adoption, community development.

Stakeholder Engagement & Information Sharing

ERF code & documentation are live and accessible





- Project not widely advertised during FY20 due to rescope; advertizing has resumed during FY21.
- One large company has expressed interest in becoming an early adopter.
- Team plans to advertise code at upcoming workshops and conferences.
- ERF team will conduct workshops and webinars with industry and academic partners.
- ERF will assemble an advisory panel to keep the team aware of emerging industry and research priorities, as well as leveragable activities and capabilities under development elsewhere.
- Coupling with AMRWind will encourage adoption by users of ExaWind codes.
- Demonstrations of computational and physical code performance will be published in appropriate widely read journals and disseminated at popular scientific and industry conferences.

Key Takeaways and Closing Remarks

Project Impact:

- Enabling multiscale atmosphere/wind plant simulations in a wide range of wind energy workflows is essential to ensure the reliability of an electrical grid dependent upon large inputs of wind energy.
- Extensibilty of simulation code to new operating environments (complex terrain, offshore) is critical for timely responses to new challenges.

Project Performance:

- Team demonstrated resilience and adaptabilty, rescoping project, progressing on code development.
- Rigorous testing of code components
- Extensive, comprehensible documentation.

Stakeholder Engagement:

- Public Github with documentation, test problems.
- Beginning advertising and interacting with stakeholders following successful construction of the code base.