Fast Equilibration of Oceanic Tracers Software

(FEOTS)

User's Guide and Technical Documentation

Joseph Schoonover, Wilbert Weijer, Jiaxu Zhang

The Fast Equilibration of Ocean Tracers Software (FEOTS) is a set of Fortran modules and programs for post-processing output from LANL's Parallel Ocean Program (POP). However, it is written so that it can be extended for use with other General Circulation Models. Tools are provided for aiding in the diagnosis of sparse matrices that capture advection and diffusion operators that are consistent with GCM discretizations. FEOTS additionally offers tools to make use of the diagnosed operators to run offline tracer models in forward or equilibration modes.

Many thanks to Wilbert Weijer and Matthew Hecht for their support on developing this software.

- Joseph Schoonover

Contents

Part I

Users Guide

1 Getting Started

1.1 System Requirements and Software Dependencies

At a minimum, you will need 750 KB of space for the FEOTS source code and a Fortran compiler. At present, FEOTS has been tested with the GNU Fortran compiler, version 4.9.3.

NetCDF

FEOTs uses the NetCDF-Fortran API libraries for handling file I/O. For downloading and installation instructions for the NetCDF-Fortran API, visit the website

http://www.unidata.ucar.edu/software/netcdf/docs/building_netcdf_fortran.html

for more information.

1.2 Installation

Currently, an online repository for FEOTS is hosted through Los Alamos National Laboratory's internally facing gitlab site (https://gitlab.lanl.gov/schoonover/FEOTS). To gain access to the repository, send an e-mail to either Joseph Schoonover (jschoonover@lanl.gov) to be given access and downloading instructions. Currently, we are in the process of open-sourcing FEOTS, at which point it will be made publicly available on GitHub.

Once you have been given access to the FEOTS repository, the source code can be cloned,

```
git clone git@gitlab.lanl.gov:schoonover/FEOTS.git
```

This creates a directory called FEOTS that has the following subdirectories :

`build/`	Contains the master makefile for FEOTS.
`doc/`	Contains this documentation.
`examples/`	Contains example configurations that are documented in Chapter 4
`src/`	Contains the source code for the FEOTS core and for interfacing with POP. Underneath this directory, a developer would add a subdirectory for interfacing with other GCMs.

Underneath the `build/` directory there is a master makefile. Within the makefile, the `LIB` and `INC` variables must be set to the library and includes paths (respectively) for your local installation of the NetCDF-C and NetCDF-Fortran API libraries.

To verify that you have configured the makefile correctly, you can run

```
make all
```

This will attempt to compile all of the binaries associated with FEOTS. If successful, you can clean up this directory with

```
make clean
```

This removes all of the `.o` and `.mod` files that were generated during compilation.

Important!

If you experience any issues with compiling the binaries, contact Joe Schoonover at jschoonover@lanl.gov or schoonover.numerics@gmail.com . With your e-mail include your makefile and any output from running `make all`.

If you have successfully compile all of the binaries, you can proceed to the Examples problems. Keep in mind that many of the example problems depend on a database of transport operators. If you do not have such a database, one needs to be generated, as outlined in the "Global Operator Diagnosis" example in Section (4.2). *Will we provide a public access database??*

2 FEOTS Workflow and Programs

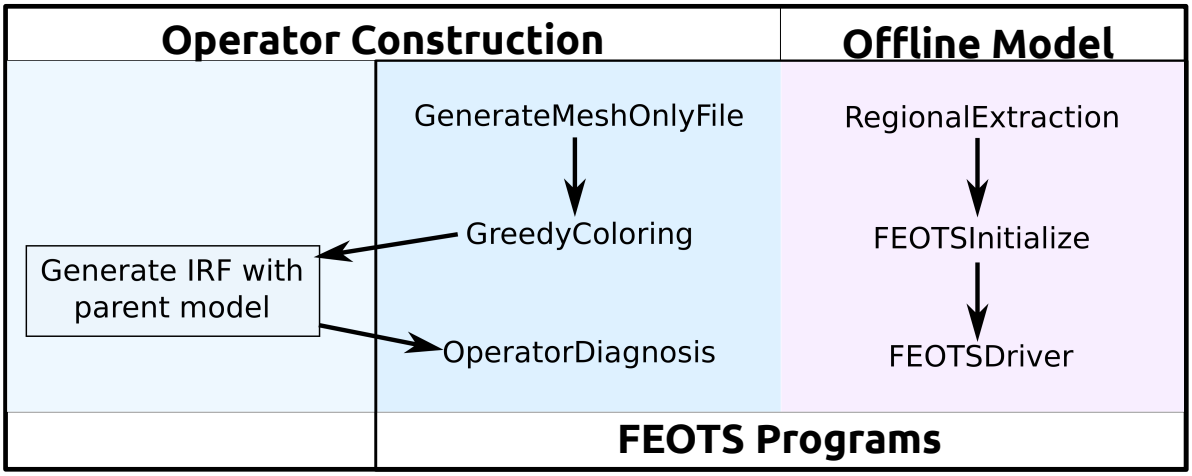

Figure 2.1: A schematic depicting the two-stage workflow of operator construction followed by the offline tracer model. To accomplish this workflow, FEOTS provides six programs for each step. The order of execution is indicated by the arrows. Prior to running the offline model, a database of transport operators must be diagnosed from a parent model.

FEOTS views the workflow for running an offline tracer model in two stages

1. Transport Operator Construction

2. Running the Offline Model

For each stage, FEOTS provides three programs that divide the workflow into incremental steps. This division of steps is purposeful and is meant to provide stopping points for the user to check for any mistakes before proceeding through the workflow.

In this section, a summary of the function of each program within the FEOTS workflow is given. Additionally, suggestions are given for

2.1 GenerateMeshOnlyFile

This program will take a netcdf file, output by POP, extract only the mesh information and write a the mesh to a netcdf file with an additional "mask" field needed for the GreedyColoring.

2.2 GreedyColoring

Generates the impulse fields given a mesh and a stencil name. Currently, only the Lax Wendroff stencil is included. The impulse fields are then written to a netcdf file for use in POP. Additionally, the adjacency graph and its coloring are written to a binary file for later use.

2.3 IRF Generation with POP

2.4 OperatorDiagnosis

Uses the Adjacency graph and its coloring (generated by GreedyColoring) in addition to impulse response fields to extract the sparse matrix representation of the transport operators. Before calling this program, GreedyColoring must be called to generate impulse fields, and the impulse fields need to be passed through POP to generate the impulse response fields

2.5 GenMask

This program can be used to generate a mask for picking out a domain for a regional simulation. It is likely that you will need to get a copy of the source code (`src/POP/programs/GenMask.f90`) and modify it to suit your particular application. This source code can be placed in any example directory that you create.

2.6 RegionalExtraction

Given latitudinal and longitudinal boundaries, this program constructs a data structure that maps between the global mesh and a regional mesh. Masks and mappings are constructed that are used to extract the appropriate rows and columns of the sparse transport matrices. These mappings are written to a binary file for later usage. A regional POP mesh is constructed and a tracermask field is filled in to indicate boundary cells; the regional POP mesh is written to a netcdf file for user inspection. Care was taken to ensure a correct mapping occurs for regions that cross the prime-meridian. Currently the regional extraction tool has not been tested on regions around the tripole centers.

2.7 ExtractOceanState

2.8 FEOTSInitialize

This program is used to set the initial conditions, source terms, relaxation time scales, additional masks, and hard-set values for the tracer fields. The initial fields and other terms are written to a netcdf file (Tracer.init.nc) for user inspection, to verify the configuration before beginning the actual run.

2.9 FEOTSDriver

The driver program manages the call to the forward integrator or the equilibration routines, and handles file I/O. This portion is currently in testing right now

3 Configuring and Running FEOTS Programs

3.1 Software Acceleration with OpenMP

If you have access to multicore architectures and a Fortran compiler with OpenMP support, the `FEOTSDriver` can be accelerated by enabling OpenMP. To build an OpenMP accelerated binary for the `FEOTSDriver`, set the environment variable `OMP` to `yes`. In a bash shell,

```
export OMP=yes
```

and in tcsh

```
setenv OMP yes
```

With this variable set, executing

```
make FEOTSDriver
```

from an example directory will enable OpenMP accelerations during compilation. Before running the executable, you must set the environment variable `OMP_NUM_THREADS` to the desired number of threads.

– *provide scaling analysis on a few architectures and with a few additional openmp thread affinity options* —

3.2 Software Acceleration MPI

If you have access to multicore architectures or a small computing cluster and a compiler with MPI capabilities, MPI can be used to parallelize FEOTS. It shoud be noted that currently, MPI is only function with the Passive Dye tracer and Settling Particulate models. The MPI implementation parallelizes the FEOTS routines by assigning each MPI rank to a single particulate. This is particularly useful for water-mass tagging problems over multiple boundaries, where the number of tracers can grow quickly.

To build the MPI flavor of the `FEOTSDriver`, set the environment variable `MPI` to `yes`. In a bash shell,

```
export MPI=yes
```

and in tcsh

```
setenv MPI yes
```

With this variable set, executing

```
make FEOTSDriver
```

from an example directory should compile the FEOTSDriver using an MPI capable compiler. If your compiler is not in your default search path, you will need to modify a section of the makefile (`FEOTS/build/makefile`), so that the variable `FC` points to the correct compiler.

```
ifeq (${MPI},yes)
   FC=mpif90
   OPT+=-DHAVE_MPI
endif
```

To run the FEOTSDriver with MPI, you must specify the number of processes to be the number of tracers plus one. This configuration is required since the master rank (rank 0) manages file I/O, while the other ranks handle computations. An incorrect configuration will cause the FEOTSDriver to stop within the initialization phase. As an example, if `nTracers=3` (and water mass tagging is turned off), then

```
mpirun -np 4 ./FEOTSDriver
```

would be the correct configuration. If water mass tagging is turned on, the number of tracers used is the number of masks multiplied by the number of layers.

3.3 Hybrid MPI-OpenMP

It is possible to run the `FEOTSDriver` with a hybrid MPI-OpenMP flavor. Before compilation, the environment variables `MPI` and `OMP` need to be set to `yes`. In a bash shell,

```
export MPI=yes
export OMP=yes
```

and in tcsh

```
setenv MPI yes
setenv OMP yes
```

With this variable set, executing

```
make FEOTSDriver
```

from an example directory should compile the `FEOTSDriver` using an MPI capable compiler with OpenMP enabled. Before running the executable, you must set the environment variable `OMP_NUM_THREADS` to the desired number of threads for each MPI rank.

The simplest method for executing the MPI-OpenMP flavor of the `FEOTSDriver` is to use (e.g.)

```
mpirun -np 4 -x OMP_NUM_THREADS ./FEOTSDriver
```

It is likely that this encantation will not yield ideal performance for any given architecture. To obtain better performance, you will want to set the layout of the ranks and threads for the specific architecture you are running on.

3.4 The Namelist File

POPMeshOptions

MeshType	
Type	`Character`
Description	Specifies the type of mesh used in the POP simulation. The mesh type is used to determine where periodic boundary conditions should be applied to aid in the construction of an adjacency graph and for performing regional mesh extraction.
Impacted Programs	`GreedyColoring`, `OperatorDiagnosis`, `RegionalExtraction`
Valid Options	"PeriodicTripole"

StencilType	
Type	`Character`
Description	Specifies the advection stencil used in the POP simulation. The stencil type is used to determine cell connectivity when building an adjacency graph
Impacted Programs	`GreedyColoring`, `OperatorDiagnosis`, `RegionalExtraction`
Valid Options	"LaxWendroff"

Regional	
Type	`Logical`
Description	A flag that specifies whether or not you are running a regional simulation. A regional domain, in FEOTS, is a domain that is a subset of the parent model. If Regional = .TRUE., you must also specify the bounding latitudes and longitudes.
Impacted Programs	`RegionalExtraction`, `FEOTSInitialize`, `FEOTSDriver`
Valid Options	.TRUE. , .FALSE.
south	
Type	`Real`
Description	Specifies the southern boundary of a regional domain in °N. Negative values indicate latitudes in the southern hemisphere, positive values indicate latitudes in the northern hemisphere.
Impacted Programs	`RegionalExtraction`
Valid Options	[-90.0,90.0], south $<$ north
north	
Type	`Real`
Description	Specifies the northern boundary of a regional domain in °N. Negative values indicate latitudes in the southern hemisphere, positive values indicate latitudes in the northern hemisphere.
Impacted Programs	`RegionalExtraction`
Valid Options	[-90.0,90.0], north $>$ south

west	
Type	`Real`
Description	Specifies the western boundary of a regional domain in °E. Negative values indicate longitudes west of the prime-meridian, positive values indicate longitudes east of the prime-meridian.
Impacted Programs	`RegionalExtraction`
Valid Options	[-360.0,360.0], west < east

east	
Type	`Real`
Description	Specifies the eastern boundary of a regional domain in °E. Negative values indicate longitudes east of the prime-meridian, positive values indicate longitudes east of the prime-meridian.
Impacted Programs	`RegionalExtraction`
Valid Options	[-360.0,360.0], east > west

MaskFile	
Type	`Character`
Description	Specifies the NetCDF file to read and write a regional mask for selecting the regional domain. When `Regional` is set to `.FALSE.` this option is ignored. When the `MaskFile` is set, the latitude and longitude bounds are ignored
Impacted Programs	`GenMask`, `RegionalExtraction`, `FEOTSInitialize`, `FEOTSDriver`
Valid Options	"LaxWendroff"

TracerModelOptions

TracerModel	
Type	`Character`
Description	Specifies the type of passive tracer model to use. This influences the conditional evaluation in `src/solutionstorage/TracerStorage_Class.f90`, subroutine `CalculateTendency_TracerStorage`. If the Radionuclide Model or Settling Model are used, the settling velocity needs to be set.
Impacted Programs	`FEOTSInitialize`, `FEOTSDriver`
Valid Options	"DyeModel", "RadionuclideModel", "SettlingModel"
WaterMassTagging	
Type	`Logical`
Description	A flag that specifies whether or not you are using the water-mass tagging capabilities in the code. If this feature is turned on and you are running a regional configuration, you must generate a database of the ocean state (temperature, salinity, density) using the `ExtractOceanState` program.
Impacted Programs	`FEOTSInitialize`, `FEOTSDriver`
Valid Options	.TRUE. , .FALSE.

settlingVelocity	
Type	`Real` (Double Precision)
Description	If `TracerModel`="RadionuclideModel" or "SettlingModel", the settling velocity is the fixed vertical velocity used to generate the vertical settling operator.
Impacted Programs	`FEOTSInitialize`, `FEOTSDriver`
Valid Options	"DyeModel", "RadionuclideModel", "SettlingModel", "Buoyant-Tracers"

nTracers	
Type	`Integer`
Description	The number of tracers for the offline model. If `TracerModel`="RadionuclideModel", this setting is ignored; the Radionuclide model supports 1 particulate and 1 radionuclide (2 tracers). For any other model, this parameter dictates how many tracer fields you want to work with. *Since FEOTS operates on a shared memory parallelism, care must be taken to ensure that you do not run out of physical memory.*
Impacted Programs	`FEOTSInitialize`, `FEOTSDriver`
Valid Options	≥ 1

runMode	
Type	`Character`
Description	Indicates whether you are running in "Forward" mode, in which transient behaviors of the tracers are desired, or "Equilibrium" mode in which an equilibrium solution is sought.
Impacted Programs	`FEOTSDriver`
Valid Options	"Forward", "Equilibrium"
timeStepScheme	
Type	`Character`
Description	Specifies the explicit integration scheme for any of the tracer models. Can be set to implement 1^{st} Order Forward Euler, 2^{nd} Order Adams-Bashforth, or 3^{rd} Order Adams Bashforth.
Impacted Programs	`FEOTSDriver`
Valid Options	"Euler", "AB2", "AB3"
dt	
Type	`Real` (Double Precision)
Description	The time step size for any forward integration. Note that this also impacts the equilibrium mode.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0.0

nTimeSteps	
Type	`Integer`
Description	The number of desired time steps for running the `FEOTSDriver` in Forward mode. In Equilibrium mode, this option is ignored.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0

nStepsPerDump	
Type	`Integer`
Description	In Forward mode, this is the number of time steps taken between each netcdf file output. In Equilibrium mode, this is the number of nonlinear iterations taken between each netcdf pickup file output.
Impacted Programs	`FEOTSDriver`
Valid Options	≥ 1

OperatorOptions

OperatorPeriod	
Type	`Real` (Double Precision)
Description	The length of time (in seconds) for which each transport operator should be applied.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0.0

nOperatorsPerCycle	
Type	`Integer`
Description	The number of transport operators to cycle over before repeating. *Ex.:* If nOperatorsPerCycle=5, FEOTS will use the first five transport operators and continue integration by repeating over these operators.
Impacted Programs	`FEOTSDriver`
Valid Options	≥ 1

FileOptions

extractRegionalOperators	
Type	`Logical`
Description	A flag used to indicate whether regional operators should be extracted. If set to .FALSE., the `RegionalExtraction` program will only generate a regional mesh and the local-to-global mapping. If set to .TRUE., the `RegionalExtraction` program will also diagnose regional transport operators.
Impacted Programs	`RegionalExtraction`
Valid Options	`.TRUE.`, `.FALSE.`

graphFile	
Type	`Character`
Description	Specifies the file where the FEOTS Adjacency graph is stored for a particular parent model.
Impacted Programs	`GreedyColoring`, `DiagnoseOperators`, `RegionalExtraction`
Valid Options	File name with path less than or equal to 400 characters.

IRFListFile	
Type	`Character`
Description	Specifies the file that contains a list of all of the Impulse Response Function NetCDF files generated by the parent model. *Ex :* If the environment variable `IRFDIR` defines the path to the IRF files, this file can be generated with the command `ls $IRFDIR > IRFfiles.txt` . Then, in the namelist file (`runtime.params`), `IRFListFile = IRFfiles.txt,` .
Impacted Programs	`DiagnoseOperators`, `RegionalExtraction`
Valid Options	File name with path less than or equal to 400 characters.

IRFStart	
Type	`Integer`
Description	Specifies the file that contains a list of all of the Impulse Response Function NetCDF files generated by the parent model. *Ex :* If the environment variable `IRFDIR` defines the path to the IRF files, this file can be generated with the command `ls $IRFDIR > IRFfiles.txt` . Then, in the namelist file (`runtime.params`), `IRFListFile = IRFfiles.txt,` .
Impacted Programs	`DiagnoseOperators`, `RegionalExtraction`
Valid Options	$0 <$ `IRFStart` $\leq$ `nIRFFiles`

nIRFFiles	
Type	`Integer`
Description	The number of Impulse Response Function NetCDF files generated by the parent model. *Ex :* If the environment variable `IRFFiles.txt` is set as the `IRFListFile` and the number of IRF files is not known a'priori, the command `wc -l IRFFiles.txt` will return the number of IRF files.
Impacted Programs	`DiagnoseOperators`, `RegionalExtraction`
Valid Options	> 0

feotsOperatorDirectory	
Type	`Character`
Description	Specifies the directory where the transport operators for a given parent model should be stored or read from.
Impacted Programs	`DiagnoseOperators`, `RegionalExtraction`, `FEOTSDriver`
Valid Options	File name with path less than or equal to 400 characters.

regionalOperatorDirectory	
Type	`Character`
Description	Specifies the directory where a set of regional transport operators should be stored or read from.
Impacted Programs	`RegionalExtraction`, `FEOTSDriver`
Valid Options	File name with path less than or equal to 400 characters.

operatorBaseName	
Type	`Character`
Description	When the transport operators are written to file, a connectivity (.conn) and data (.data) file are written for advection and difusion operators with the same "base-name". `operatorBaseName` determines this base name. *Ex :* If `operatorBaseName = pop_03_tripole,` then the transport operator files will be written in files `pop_03_tripole_advect.NNNNN.data`, `pop_03_tripole_advect.NNNNN.conn`, `pop_03_tripole_diffu.NNNNN.data`, and `pop_03_tripole_diffu.NNNNN.conn`, where the `NNNNN` is an integer padded with zeros indicating which operator it corresponds to.
Impacted Programs	`DiagnoseOperators`, `RegionalExtraction`, `FEOTSDriver`
Valid Options	String less than or equal to 100 characters.

meshFile	
Type	`Character`
Description	Specifies the NetCDF file that stores the POP Mesh information associated with the full parent model in addition to a mask generated from the KMT field.
Impacted Programs	`GenerateMeshOnlyFile`, `DiagnoseOperators`, `RegionalExtraction`, `FEOTSInitialize`, `FEOTSDriver`
Valid Options	File name with path less than or equal to 400 characters.

regionalMeshFile	
Type	`Character`
Description	Specifies the NetCDF file that stores a regional mesh with its mask for a given parent model and regional configuration.
Impacted Programs	`RegionalExtraction`, `FEOTSInitialize`, `FEOTSDriver`
Valid Options	File name with path less than or equal to 400 characters.

JFNKOptions

maxItersJFNK	
Type	`Integer`
Description	Specifies the maximum number of outer, nonlinear, iterations for the "Newton part" of the JFNK solver.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0

toleranceJFNK	
Type	`Real` (Double Precision)
Description	The error tolerance (stop criteria) for the nonlinear solver.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0.0 , *Default :* 10^{-7}

maxItersGMRES	
Type	`Integer`
Description	Specifies the maximum number of restarts for the GMRES-with-restarts inner, linear, solver; the "Krylov" part of the JNFK solver.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0

mInnerItersGMRES	
Type	`Integer`
Description	Specifies the size of the Krylov subspace to search for a solution on the inner iterates of the GMRES solver.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0

toleranceGMRES	
Type	`Real` (Double Precision)
Description	The error tolerance (stop criteria) for the GMRES linear solver.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0.0 , *Default :* 10^{-7}

JacobianStepSize	
Type	`Real` (Double Precision)
Description	The step size for approximating the Jacobian matrix action.
Impacted Programs	`FEOTSDriver`
Valid Options	> 0.0, *Default :* 10^{-2}

isPickupRun	
Type	`Logical`
Description	Indicates whether the equilibration run is a pickup run. If set to `.TRUE.`, the `FEOTSDriver` will attempt to open a file `Tracer.pickup.nc` as the initial condition. If set to `.FALSE.`, the `FEOTSDriver` will attempt to open a file `Tracer.init.nc` as the initial condition.
Impacted Programs	`FEOTSDriver`
Valid Options	`.TRUE.`, `.FALSE.`

3.5 Setting up your own example

When developing your own example problem for the offline tracer model, it is easiest to copy an existing example that is most similar to your desired setup, and make modifications from there. In any situation, you will want to copy the following files from a pre-configured example :

`FEOTSInitialize.f90`	Fortran source code for setting up the initial conditions, source terms, masks, and hard-set tracer values.
`GenMask.f90`	An optional Fortran source code for generating a regional mask.
`makefile`	A local makefile that manages calls to the master makefile and assures compilation of the local FEOTSInitialize.f90 source code for the FEOTSInitialize program.
`runtime.params`	Namelist file for configuring your model simulation

Generating Impulse Fields for IRF Generation

Disclaimer *: Currently, FEOTS is only setup for working with the POP Tripole Grid and the Lax-Wendroff advection scheme (7 point stencil). However, adding other grids and advection schemes is quite easy. Contact schoonover.numerics@gmail.com about adding more functionality.*

Setting your File I/O Options

Regional Simulation with Lat/Lon Bounds

Regional domains and transport operators can be constructed using one of two strategies :

1. Specifying latitude and longitude boundaries
2. Providing a mask in a NetCDF file

In either approach, the `Regional` parameter underneath the `POPMeshOptions` namelist must be set to `.TRUE.` .

The simplest method for performing a regional simulation is to specify the latitude and longitude boundaries using the `south`, `north`, `west`, and `east` parameters. In this case, the `POPMeshOptions` namelist within the `runtime.params` namelist file would look similar to the example given below

```
&POPMeshOptions
Regional = .TRUE.,
south    = -60.0,
north    = -20.0,
west     = -10.0,
east     = -20.0,
/
```

With this configuration, in addition to the File Options being configured properly, you can compile and run the `RegionalExtraction` program to generate the necessary files for

performing a regional experiment. By default, this program will generate a NetCDF file that contains the regional mesh and will subsample all of the transport operators. If you need to fine-tune the regional extraction and temporarily avoid extracting all of the operators, you can set the `extractRegionalOperators` parameter to `.FALSE.` under the `FileOptions` namelist. In this mode, the `RegionalExtraction` program will only create the NetCDF file corresponding to the name provided in the `regionalMeshFile` parameter.

Regional Simulation with a User-provided Mask

Initial Conditions

Of these three files, only FEOTSInitialize.f90 and runtime.params need to be modified to design a new example, provided you have access to an existing set of transport operators. Within FEOTSInitialize.f90, there is one subroutine that can be modified to set up the initial tracer distribution and the source and mask terms : `InitialConditions`

To explain how to assign the initial conditions, a brief overview of the `POP_FEOTS` data structure is needed.

Water Mass Tagging

Disclaimer *: Currently, water mass tagging is configured to work only with regional experiments. Enabling water mass tagging on a global simulation has not been tested.*

The development of an experiment with water mass taggging begins at the `RegionalExtraction` stage of the FEOTS workflow. If your region is chosen without providing a region mask, then all of the boundary points by default are set as prescribed boundary points. This means that water mass tagging will occur at all boundary points. Alternatively, if you provide a region mask, setting grid points to -1 corresponds to that point being a prescribed point. When performing water mass tagging experiments, it is recommended that you supply a region mask as this provides you with the most control on where to prescribe tracer injections.

In addition to setting up the regional domain and performing regional operator extraction, you must compile and run the `ExtractOceanState` program to generate a database of NetCDF files that contain the temperature, salinity, and potential density output from

POP for your region. Successful execution of the `ExtractOceanState` program should result in `Ocean.*.nc` files in the directory specified by `regionalOperatorDirectory` parameter in the `runtime.params` namelist file. At this point, you can run the `FEOTSInitialize` program.

Before running the `FEOTSDriver` program, there are a couple options in the namelist file that need to be set. Under the `TracerModelOptions` namelist, the `WaterMassTagging` parameter must be set to `.TRUE.` . It is recommended that the `TracerModel` is set to `"DyeModel"`. If you want to tag multiple water masses, set `nTracers` to the desired number. Keep in mind that increasing the number of tracers increases memory consumption and can increase wall-time.

Last, you must create a file called `watermass.config`. The first line of this file should indicate the ocean state variable that you are using to tag water masses. Valid options include `Temperature`, `Salinity`, or `Density`. This option is not case sensitive, though incorrect spelling or any other invalid option will result in the `FEOTSDriver` stopping. The next few lines should list the lower and upper bounds of the ocean state variable for each tracer. For example, if the contents of `watermass.config` are

```
Temperature
5.0 10.0
10.0 15.0
15.0 20.0
```

and `nTracers = 3,` in `runtime.params`, then for each operator the `FEOTSDriver` will hard set the first dye tracer to 1 where temperature values are between 5° C and 10° C and 0 at other prescribed points, the second dye tracer to 1 where temperature values are between 10° C and 15° C and 0 at other prescribed points, etc.

4 Examples

4.1 Parent Models

POP03T

Need documentation on the 0.3 degree POP Tripole configuration

4.2 Global Operator Diagnosis

`FEOTS/examples/POP03T_OperatorDiagnosis`

This example summarizes the procedure for diagnosing transport operators for use with the FEOTS Offline tracer models; it demonstrates the first stage of the FEOTS workflow.

Generating the Impulse Fields

In the first step towards generating the transport operators, the POP model should be configured and run for a single time step to produce a NetCDF file that contains the mesh of the parent model. In this example, the parent model is POP03T, which is described in Section 4.1. The contents of the `runtime.params` namelist file is shown here

```
&POPMeshOptions
MeshType    = 'PeriodicTripole',
StencilType = 'LaxWendroff',
/
&TracerModelOptions
```

```
/
&OperatorOptions
/
&FileOptions
meshfile                  ='/usr/projects/cesm/FastSolver/feots/
                            database/POP_0.3_Operators_5DayAvg/POP_03deg_mesh.nc',
graphfile                 ='/usr/projects/cesm/FastSolver/feots/
                            database/POP_0.3_Operators_5DayAvg/
                            pop_03_periodic-tripole_laxwendroff',
operatorBaseName          = 'pop_03_periodic-tripole',
feotsOperatorDirectory    ='/usr/projects/cesm/FastSolver/feots/
                            database/POP_0.3_Operators_5DayAvg/Global/',
IRFListFile               = 'IRFList_5dayAvg.txt',
IRFStart                  = 1,
nIRFFiles                 = 365,
/
&JFNKOptions
/
```

The `MeshType` is set to `PeriodicTripole` and the `StencilType` is set to LaxWendroff, consistent with the configuration of the POP03T parent model. The `FileOptions` namelist determines where the read the parent model mesh from, and where to write the adjacencey graph file and the transport operators. A description of each of the options in the namelist file are given in Sections 3.4 and 3.4.

To generate the impulse fields that will be passed to the POP model, the `GreedyColoring` program is used. Underneath the directory `FEOTS/examples/POP03T_OperatorDiagnosis`, run

```
make GreedyColoring
```

This compiles source code for generating the executable for generating the Impulse Fields. This program is run by executing

```
./GreedyColoring
```

The output from running this program for the POP03T parent model is shown in Figure

Figure 4.1: This image shows the dye source on the southeast coast of Africa...

4.3 Agulhas Regional Operator Diagnosis and Passive Dye Injection

`FEOTS/examples/agulhas`

In this example, the equation

$$c_t + \vec{u} \cdot \nabla c = r(c_f - c) \tag{4.1}$$

is solved, where c is the concentration of a passive "dye" tracer, $\vec{u}$ is the velocity field, c_f is a relaxation field, and r is a spatially dependent relaxation frequency. For this example, the relaxation field and frequency are specified as a Gaussian in latitude and longitude and

Table 4.1:

c_0	5.0
r_0	2.31×10^{-5} s^{-1}
x_0	$31.5°E$
y_0	$31.0°S$
L	$0.354°$

independent of the vertical coordinate z,

$$c_f = c_0 \tag{4.2a}$$

$$r = r_0 e^{-\left(\frac{(x-x_0)^2+(y-y_0)^2}{2L^2}\right)} \tag{4.2b}$$

where the parameters c_0, r_0, x_0, y_0, and L are shown in Table 4.1 and the dye source field at the surface layer is shown in Figure 4.1.

The advection is modeled using one year repeat cycles of the 5-day averaged transport operators diagnosed from the POP03T parent model; the advection operator is diagnosed from the 2nd order Lax Wendroff advection scheme.

Configuration

The setup for this model can be described by the contents of the `runtime.params` namelist file. Here, the impact of each option in the namelists are explained. The first namelist is `POPMeshOptions` and is shown below.

```
&POPMeshOptions
MeshType    = 'PeriodicTripole',
StencilType = 'LaxWendroff',
Regional    = .TRUE.,
south       = -60.0,
east        = 50.0,
west        = -10.0,
north       = -20.0,
/
```

The mesh-type is specified as a periodic tripole mesh, to correspond to the type of mesh that is used in the POP03T parent model. The stencil-type is specifies that the advection scheme is the second order Lax-Wendroff. These options only influence the `GreedyColoring` and the `OperatorDiagnosis` programs; they do not influence the initialization and driver programs.

The "Regional" flag is set to `.TRUE.` to indicate that we are running a regional simulation, and the latitudinal and longitudinal boundaries are set encompass the southern tip of Africa and the Agulhas retroflection. When a regional simulation is desired, it is necessary to run the `RegionalExtraction` program first in order to build a database of transport operators for the specified region.

Regional Operator Extraction

When extracting regional operators, you must have global operators already diagnosed from the parent model. The `FileOptions` namelist is used to set up the necessary parameters for the `RegionalExtraction` program. Here, the relevant options are shown and discussed

```
&FileOptions
extractRegionalOperators  = .TRUE.,
meshfile                  ='/usr/projects/cesm/FastSolver/feots/database/
                            POP_0.3_Operators_5DayAvg/POP_03deg_mesh.nc',
regionalmeshfile          ='Agulhas_mesh.nc',
graphfile                 ='/usr/projects/cesm/FastSolver/feots/database/
                            POP_0.3_Operators_5DayAvg/
                            pop_03_periodic-tripole_laxwendroff',
operatorBaseName          = 'pop_03_periodic-tripole',
feotsOperatorDirectory    ='/usr/projects/cesm/FastSolver/feots/database/
                            POP_0.3_Operators_5DayAvg/Global/',
regionalOperatorDirectory ='/usr/projects/cesm/FastSolver/feots/database/
                            POP_0.3_Operators_5DayAvg/Agulhas/',
/
```

The `extractRegionalOperators` flag is set to `.TRUE.` to indicate that we want to extract regional operators. The `meshfile` points to the location of a NetCDF containing the global mesh for the parent model. The `regionalmeshfile` points to the desired location to write out the regional mesh after it has been extracted from the global mesh. The `graphfile` points to the location of the FEOTS generated graph-file associated with parent model's global mesh and advection scheme; this was generated with the `GreedyColoring` program. `feotsOperatorDirectory` points to the directory that contains the global transport operators and `regionalOperatorDirectory` points to a directory where you will write the regional operators to hard disk.

To build the RegionalExtraction executable, run

```
make RegionalExtraction
```

Then, to run this executable

```
./RegionalExtration
```

Running the Model

After the regional operators have been extracted, the namelists `TracerModelOptions` and `OperatorOptions` must be set up to run the initialization and driver programs. The settings for these namelists are shown here :

```
&TracerModelOptions
TracerModel = 'DyeModel',
dt          = 1080.0,
runMode     = 'Forward',
nStepsPerDump   = 80,
iterInit        = 0,
nTimeSteps      = 160,
/
&OperatorOptions
operatorPeriod = 432000.0,
```

```
nOperatorsPerCycle = 73,
/
```

In the `TracerModelOptions`, it is indicated that the tracer model is the Passive Dye Model. The simulation will proceed in Forward Mode, so that we can model the transient behavior of the dye spreading from the source. The integration uses a time step of 1080 seconds and every 80 time steps, an output file is written. Currently the configuration shows only 160 time steps to be done. The `OperatorOptions` indicate that each operator should be applied for intervals of 432,000 seconds (5 days) and we will use repeat cycles of 73 operators (1 year).

To run the model, initial conditions must first be generated. This is then followed by the driver program

```
make FEOTSInitialize FEOTSDriver
./FEOTSInitialize
./FEOTSDriver
```

With the given configuration, this should generate four NetCDF files : Tracer.init.nc, Tracer.0000000000.nc, Tracer.0000000080.nc, Tracer.0000000160.nc.

4.4 Global Particulate-Radionuclide

Part II

Technical Documentation

5 Introduction

Biological and chemical tracers can be transported throughout the world's oceans through advection and diffusion processes. Additionally, these tracers can change their concentration/activity levels through coupled interactions that are often nonlinear. The time scale for equilibration of most oceanic passive[1] tracers is on the order of hundreds to thousands of years. Global simulations of the ocean are now routinely being conducted at resolutions of 10 km and smaller, for which the explicit advective CFL limit yields a time step on the order of $\frac{1}{10}$ day. Online simulations, in which tracers are advected in tandem with forward integration of the fluid momentum equations, will typically have even more restrictive stability constraints due to the presence of other, faster modes in the momentum equations. Offline simulations do not suffer from the additional overhead associated with a full-blown GCM, though a tremendous hurdle remains in using the velocity and diffusivity fields in a manner consistent with GCM discretization and parameterizations. The Fast Equilibration of Ocean Tracers Software (FEOTS) offers the ability to overcome this hurdle by providing tools to

1. Aid in the diagnosis of advection and diffusion operators from online fluid simulations,
2. Perform offline forward integration of passive tracers, and
3. Rapidly equilibrate passive tracer systems through root-finding and minimization strategies.

FEOTS is inspired by the work of those before us, particularly Bardin et al. (2014), Primeau (2005) and Khatiwala et al. (2005), and has been guided into existence through

[1] *Passive, here, means that the tracer does not impact the momentum of mass balances of the fluid's governing equations*

communications with Francois Primeau (UC Irvine,) Anne Bardin (UC Irvine), and Keith Lindsay (NCAR). FEOTS currently has an interface to work with the Parallel Ocean Program (POP), but has been designed to remain rather agnostic to choice in General Circulation Model. In this manual, our view of passive tracer systems are described along with a generic strategy for diagnosing transport operators online and applying them offline are given. The tools provided for working with POP are described to clearly illustrate the FEOTS workflow.

6 Methods

6.1 Offline Forward Integration

Volume Correction

6.2 Equilibration Techniques

The motivating problem for the offline tracer model is to obtain an "equilibrated" solution to

$$\vec{c}_t + \nabla \cdot (\vec{u}(\vec{x}, t)\vec{c}) = \vec{s}(\vec{c}, t) + \vec{f}_{mix}(\vec{c}, t), \tag{6.1}$$

where $\vec{c}$ is a vector of passive tracers, $\vec{u}$ is the fluid velocity field, $\vec{s}$ is any source, sink, or tracer coupling terms, and $\vec{f}_{mix}$ is a representation of unresolved mixing. When spatially discretized, (6.1) becomes

$$\frac{d\vec{C}}{dt} = \boldsymbol{A}(t)\vec{C} + \vec{S}(\vec{C}, t) \tag{6.2}$$

where $\vec{C}$ is a vector of discrete tracer values (a vector of vectors), $\boldsymbol{A}(t)$ is the time-depend matrix that represents the advection and mixing operators, and $\vec{S}$ is the discretized source/sink or coupling terms.

Fixed Point Problem with JFNK

For a steady flow in the absence of time dependent sources or sinks, steady state solutions to (6.1) are well-defined. So long as there is time dependence present in any of these operators, a true steady state solution for the tracers does not exist. Primeau (2005) outlined

a problem formulation that has proven to be successful for coarse, non-eddying, resolution ocean simulations; this approach is outlined in the next section.

Given an initial condition for the discretized tracer conservation laws, (6.2) can be integrated over a time interval from $t = t_0$ to $t = t_0 + T$, which gives

$$\vec{c}(t_0 + T) = \vec{c}(t_0) + \int_{t_0}^{t_0+T} (\boldsymbol{A}(t)\vec{C} + \vec{S}(\vec{C}, t))\, dt. \tag{6.3}$$

One can view this time integration as a map that transforms $\vec{c}(t_0)$ into $\vec{c}(t_0 + T)$,ie,

$$\vec{M}(\vec{c}_0) = \vec{c}(t_0) + \int_{t_0}^{t_0+T} (\boldsymbol{A}(t)\vec{C} + \vec{S}(\vec{C}, t))\, dt. \tag{6.4}$$

Primeau suggested that an equilibrium solution to (6.2) is defined as the fixed point of the map. It is the initial condition that, when integrated over the interval $[t_0, t_0 + T]$, returns back to itself. In this formulation, the problem is stated as follows :

Find the tracer state $\vec{c}_0$ *s.t.*

$$\vec{M}(\vec{c}_0) = \vec{c}_0 \tag{6.5}$$

In general, the mapping function $\vec{M}$ is nonlinear in the tracer state. In this case, it has been common practice to use the Jacobian-Free Newton Krylov (JFNK) method to approximate solutions to (6.5). However, if the source and coupling terms are linear in the tracer state, then (6.5) is linear in the tracer field and can be solved approximately with an iterative solver (e.g. GMRES, BiCGStab).

Galerkin Minimization

6.3 Operator Diagnosis

7 Supported Tracer Models

7.1 Dye Model

$$c_t + \vec{u} \cdot \nabla c = r(c_f - c) \tag{7.1}$$

7.2 Settling Tracers

$$c_t + \vec{u} \cdot \nabla c - w_s c_z = r(c_f - c) \tag{7.2}$$

7.3 Particulate-Radionuclide Pair

In this section, the equations for modeling naturally occuring radionuclides in the oceans. This simplified models is motivated by the desire to better understand the relationship of sedimentary $^{231}Pa/^{230}Th$ distributions as a function of

1. Particulate flux distributions
2. Particulate settling velocities
3. Particulate concentrations and scavenging efficiencies
4. Ocean circulation strength and patterns

However, this model can be applied to any pair of settling particulate and scavenged radionuclide pairs.

In this model, particulates are modelled by a concentration field (with units of $\frac{mass\ particles}{mass\ fluid}$). The particulates are assumed to have a constant vertical settling velocity given by w_s. Additionally, they can be transported through the fluid through advective and diffusive processes. To model the particulate field we use the conservation law,

$$P_t + \underbrace{\nabla \cdot (\vec{u}P - \boldsymbol{\kappa}\nabla P)}_{OceanCirculation} - \underbrace{w_s P_z}_{Settling} = 0 \tag{7.3}$$

where P is the particle concentration field and $\vec{u}$ and $\boldsymbol{\kappa}$ are the velocity field diffusivity tensor (respectively) associated with the ocean circulation.

^{231}Pa and ^{230}Th are produced by the radioactive decay of ^{235}U and ^{234}U respectively. Uranium activities have been shown to be consistent with a steady state distribution (**?**), and have often been regarded as uniform in the ocean (**?**). ^{235}U comprises 0.711% of all naturally occuring Uranium and has a half-life of (approximately) 703.8 Myr. In contrast, ^{234}U, a decay product of ^{238}U accounts for only 0.0055% of all naturally occuring Uranium and has a relatively short half-life of 2.245×10^{-1} Myr. ^{231}Pa and ^{230}Th exhibit radioactive decay to ^{227}Ac and ^{226}Ra respectively and have half-lives of 32.76 Kyr and 75.38 Kyr respectively. The difference in the half-lives and concentrations of the parent Uranium isotopes and the radioactive decay of Protactinium and Thorium results in a net production ratio (Protactinium to Thorium) of 0.0926 (**?**).

Being suspended in the ocean, Protactinium and Thorium are also transported by the ocean through advective and diffusive processes. These radionuclides are also "particle-reactive". They can be adsorbed onto particulates (e.g. biogenic opal, carbonate) and transported into marine sediments as the particulates settle out of the water column. This process is known as "scavenging". The scavenging flux is modeled (here and in **?**) as the product of the particulate settling velocity and the fraction of radionuclide that adsorbs onto the particulates,

$$F_{sc} = -w_s R_p. \tag{7.4}$$

The particle settling velocity is w_s and R_p is the fraction of particulate associated radionuclide. This fraction is calculated using two rules :

1. The total radionuclide activity is the sum of a particulate associated fraction and the

dissolved (R_d) fraction,

$$R = R_p + R_d. \tag{7.5}$$

2. The ratio of particulate associated to dissolved radionuclide is proportional to the particle concentration,

$$\frac{R_p}{R_d} = kP. \tag{7.6}$$

This second rule, given by (7.6), is called the ***Equilibrium-Scavenging model*** and k is the "scavenging efficiency coefficient". Combining (7.5) and (7.6) allows us to write the scavenging flux in terms of the total radionuclide concentration

$$F_{sc} = -\frac{kP}{1+kP} w_s R. \tag{7.7}$$

The conservation law for a scavenged radionuclide is then

$$R_t + \underbrace{\nabla \cdot (\vec{u}R - \boldsymbol{\kappa}\nabla R)}_{Ocean Circulation} - \underbrace{\left(w_s \frac{kP}{1+kP} R \right)_z}_{Scavenging} = Q, \tag{7.8}$$

where Q is the uniform and constant source of the radionuclide activity from the net radioactive decay.

The single pair particulate-radionuclide equations are

$$P_t + \nabla \cdot (\vec{u}P - \boldsymbol{\kappa}\nabla P) - w_s P_z = 0 \tag{7.9a}$$

$$R_t + \nabla \cdot (\vec{u}R - \boldsymbol{\kappa}\nabla R) - \left(\frac{kP}{1+kP} w_s R \right)_z = Q \tag{7.9b}$$

7.4 Buoyant Tracers

In this section, the equations for a buoyant passive tracer are derived. This model is motivated by the need to predict the locations of high microplastic concentrations in the oceans.

In addition to the background transport by the ocean circulation, the vertical motion of a constant volume particle of density ρ_p in a fluid of density ρ is governed by

$$(w_p)_t = \underbrace{-\frac{(\rho_p - \rho)g}{\rho_p}}_{buoyancy} - \underbrace{C_d w_p}_{Drag} \tag{7.10}$$

where w_p is the vertical velocity of the particle. For a large collection of particles that are modeled as a continuum of noninteracting particle, the Eulerian velocity is equivalent to the particle velocity. Additionally, if it is assumed that the particle velocities are at equilibrium, then the vertical velocity can be determined as a balance between the particle drag and the buoyancy forces,

$$w_p = -\frac{(\rho_p - \rho)g}{\rho_p C_d}. \tag{7.11}$$

The equation governing a continuum of buoyant particles is then

$$c_t + \nabla \cdot (\vec{u}c - \boldsymbol{\kappa}\nabla c) + w_p c_z = s(x, y, z, t) \tag{7.12}$$

where s is a non-conservative source/sink term and w_p is diagnosed in (7.11).

Bibliography

A. Bardin, F. Primeau, and K. Lindsay. An offline implicit solver for simulating prebomb radiocarbon. *Ocean Modelling*, 73:45–58, 2014.

S. Khatiwala, M. Visbeck, and M.A. Cane. Accelerated simulation of passive tracers in ocean circulation models. *Ocean Modelling*, 9:51–69, 2005.

F. Primeau. Characterizing transport between the surface mixed layer and the ocean interior with a forward and adjoint global ocean transport model. *J. Phys. Oceanogr.*, 35:545–564, 2005.