Fast Equilibration of Oceanic Tracers Software

(FEOTS)

User's Guide and Technical Documentation

Joseph Schoonover

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

At a minimum, you will need xxx MB 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.

1.2 Software Dependencies

NetCDF

FEOTs uses the NetCDF-Fortran libraries for handling file I/O.

1.3 Obtaining the Code

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.

1.4 Installation

2 FEOTs Workflow

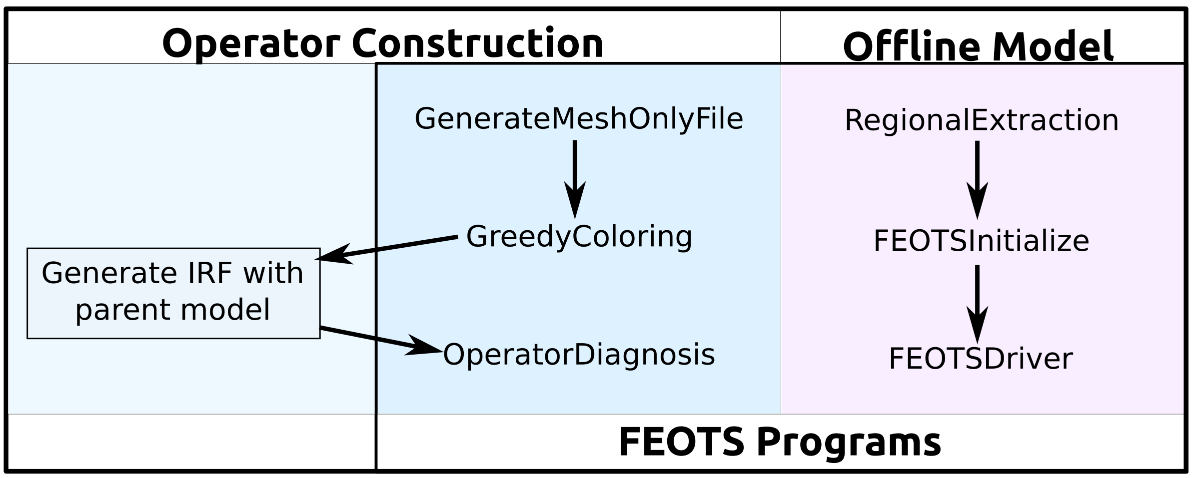

Figure 2.1: A schematic depicting the two-stage workflow of operator construction followed by the offline tracer model. To accomplish this workflow, FEOTS provides six programs for each step. The order of execution is indicated by the arrows. Prior to running the offline model, a database of transport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

2.6 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.

2.7 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 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

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"

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", "BuoyantTracers"

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"

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

mInnerItersJFNK	
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.`

4 Examples

4.1 Parent Models

POP03T

4.2 Global Operator Diagnosis

The `IRFListFile` is a file that lists all of the NetCDF files containing the impulse response functions generated by the parent model. This file is easily generated by using the `ls` command, e.g. :

```
ls /usr/projects/cesm/FastSolver/t32_IRFs/hist2/*nc > IRFList_5dayAvg.txt
```

4.3 Global Passive Dye Injections

4.4 Agulhas Regional Operator Diagnosis and Passive Dye Injection

`/feots_lite/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,

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

the relaxation field and frequency are specified as a Gaussian in latitude and longitude and independent of the vertical coordinate z,

$$c_f = c_0 e^{-\left(\frac{(x-x_0)^2+(y-y_0)^2}{2L^2}\right)} \tag{4.2a}$$

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

$$\tag{4.2c}$$

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.

Table 4.1:

c_0	5.0
r_0	
x_0	
y_0	
L	

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.

Running the Model

4.5 Passive Micro-plastics

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

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

7.2 Settling Tracers

7.3 Particulate-Radionuclide Pair

7.4 Buoyant Tracers

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.