KEI - The Tutorial

Revision History:
2014-09-29 Created Benjamin Saenz

Table of Contents

Introduction

The KPP-Ecosystem-Ice, or KEI, model consists of vertical water column model, using the KPP boundary layer mixing scheme, fully coupled to a thermodynamic sea ice model. KEI is forced at the atmospheric boundary by time-varying atmospheric data through various standard exchange energy and mass and flux parameterizations. Various unphysical assimilation routines and 'hacks' are available for developing tests for particular problems.

KEI is written in FORTRAN 95. Many helper routines for creating, editing, and post-processing data are available for MATLAB, however MATLAB is not required for run the model.

This model derives from:

- Large et al. [1994], Doney et al. [1996](KPP mixing)
- Ukita and Martinson [2001](Mixed layer - ice interactions)
- Saenz and Arrigo [2012 and 2014] (SIESTA sea ice model)
- Hunke and Lipscomb [2008] (CICE v4 ice model – various components)
- Moore et al. [2002,2004] (Biogeochemical Ecosystem Model)

KEI run files

Three things are required for KEI to run:

1) Executable (from FORTRAN compiler)
2) Run Options File
3) Forcing Data File (NetCDF)

Building the executable

There are five necessary components to build KEI: A FORTRAN 95 compiler, LAPACK, NetCDF, Make, and the KEI source code.

FORTRAN:
For Macintosh: http://hpc.sf.net (use GCC 4.8; 4.9 was broken last I checked and 5.0 is untried)
For Linux: gfortran, or intel ifort (both free for non-commercial use)
CPP or an equivalent preprocessor is also required, but is installed with GCC and/or ifort

LAPACK:
On Macintosh, LAPACK is available as a part of Developer Tools, but you must build the interface. Linking is available in the Makefile. On Linux, Ifort ships with LAPACK; heinous linking is already done in the Makefile for ifort 11.1, but for

different versions it can be tough to figure out. For gfortran, you must download LAPACK, build it and link it.

NetCDF:
On Macintosh, it must be downloaded and built with FORTRAN 90 interfaces. On linux, your distribution likely has a package you can install and link. This can be a pain.

KEI source code:
The FORTRAN 95 source code consists of ~35 files. The file that contains the Main driver routine is KEI.f90. The Makefile used by Make to build the executable should be distributed with the source code.

Make/Makefile:
If developer tools are installed, you should have the make command available. Edit the file "makefile" to suit your system. Setting the FC variable to either ifort for gfortran will give you independent control over those two compilers. You will have to edit the paths to the NetCDF include and lib directories, and also amend the linking to some version of LAPACK. (Note that ifort and MacOS X ship with LAPACK - you must build the LAPACK FORTRAN interfaces before using however). Windows is not supported, mostly because it is incredibly difficult to get the FORTRAN NetCDF interface installed on windows.

Type "`make clean`", then "`make`." The executable KEI.run should be produced. Various warnings are produced by different compilers; they seem to be benign on my test systems thus far.

Run Options File

Currently much of this file is legacy garbage, but the model is expecting some of it until I clean it up. We are interested in only lines:

Line 13: start time (in days) - corresponds to the "time" forcing variable; the number of days after the start of forcing data
Line 14: time step start (always 0), end (# desired steps), and time step length (seconds - 1hr (3600s) time step is the only one tested!).
Line 21: The PATH where output data will be written.
Line 24: Toggles for switching on/off parts of the model, such as ice, ecosystem, KPP, etc. I have only tested toggling the ecosystem and ice.
Line 50: Path/filename to where the forcing NetCDF file is located

Example run options file: run.00.so

Forcing File

The forcing file expected by KEI is a NetCDF file, with specific variables describing the grid, physical and ecosystem initializations, and atmospheric and ocean time series forcing data. The data file can be generated using MATLAB from a specific MATLAB data structure that is relatively easy to manipulate. Furthermore, the forcing and initialization data can be (mostly) generated from raw data, in the form of ECMWF ERA Interim Climatology and various NCEP BCG model outputs.

Example NetCDF forcing file: kf_200_100_2000.nc

Corresponding MATLAB structure containing the same data: kf_200_100_2000.mat

Example command to write the MATLAB structure to netcdf:

```
kei_write_forcing(kf_200_100_2000,’kf_200_100_2000.nc’)
```

Helper function examples:

Get the latitude/longitude of LTER grid points

```
[lat, lon] = grid2llLter(200,100);
```

Find the index of a CTD cast:

```
find_LTER_cast(castAll, 2000, 200, 100);
```

Retrieve and interpolate atmospheric forcing data:

```
f=kei_fdat_prepare(lat,lon,2001,1,2003,365,'/data/ECMWF_int/',0)
```

kei_prepare_eco_init.m

```
eco_init = kei_prepare_eco_init(
'GECO.IAF.20th.x1.CESM1.001.pop.h.subset.228-238.nc4',lat,lon,1:400,1)
```

Running the model

Running the KEI.run executable is simple:

```
KEI.run < options_file
```

Where options_file is the file created for the run, and which points to the data directory where the forcing file should be and the output NetCDF file will be written. Run times will vary based upon the processor speed, disk speed, compiler and optimization. KEI writes out a tremendous amount of output data, both to screen and to disk (~750 Mb/yr). Run times can be decreased by using an SSD drive for writing and reading, and by suppressing output printing by piping screen output to NULL like this:

```
KEI.run < options_file > /dev/null
```

Postprocessing and viewing model output

The majority of output NetCDF variables can be read in MATLAB for analysis using:

```
k=kei_read(‘output.nc’)
```

This can be extremely slow, especially with all BCG tracers. For most current plot commands, kei_read_plotting works faster by not reading BCG data.

Example data loading and plotting:

```
k=kei_read_plotting('john.nc')
load kf_200_100_2000; kf = kf_200_100_2000
kei_plot_fluxes(k,200,[1,length(k.time)],kf,19.625*24,2000)
kei_plot_ecoline(k,200,[1,length(k.time)],kf_200_100_2000,19.625*24)
kei_plot_ecoiceline(k,200,[1,length(k.time)],kf,19.625*24,2000)
kei_compare_ctd_plus_ice(k,castAll,2000,200,100,kf,200)
```

For quick and dirty views of physics and BGC:

```
k=kei_read(‘output.nc’)
kei_plot_ice_d(k,200)
kei_plot_eco_d(k,200)
```

kei_hacks.f90

This file contains the toggles necessary to examine certain physical and BCG processes. These are hard-coded changes, meaning the code must be re-compiled for them to take effect.

These ‘hacks’ and switches remain to be fully documented, but here is what a few of them do:

`ic_conform = 1`	Enable SSM/I sea ice concentration conforming
`fe_multiplier = 1.0`	Multiply initial Fe profile
`fe_offset = 200.`	Shift downward Fe profile(m) [negative to use forcing]
`bio_offset = 300.`	Shift living things downward (m) [0.0 to use forcing]
`ice_diatChl = 25.6/0.6`	Melting ice contains diatoms (mmol/m^3)
`ice_fe = 6.`	Melting ice releases Fe (nm)

KEI Palmer LTER mega commands

```
kei(kei([1997:2010],'run_name',1);
kei_moorings(2007:2011,run_name,1,[0,1,2]);
```

See README_KEI_Palmer_LTER.txt