Documentation for 3D K Profile Parameterization model Fortran 77 Version

Nicholas P. Klingaman
Department of Meteorology, University of Reading
E-mail: n.p.klingaman@reading.ac.uk

Accompanying Release 2, Modified October 7, 2009

1 Introduction

1.1 Model description

The three-dimensional K Profile Parameterization boundary-layer ocean model (hereafter "the 3D KPP model") is based on the KPP boundary-layer ocean model described in Large et al. (1994). While the Large et al. model runs as a single, one-dimensional water column, the 3D KPP model includes "wrappers" that allow for a horizontal matrix of water columns. It is important to note immediately that the 3D KPP model is really only quasi-3D, as there is no communication between the water columns. The model does not represent processes such as horizontal advection or upwelling from below the limited vertical domain.

Currently, the 3D KPP model may be run in forced mode—in which the model is driven by user-prescribed surface forcing data—or in coupled model—in which the model exchanges surface fields with an atmospheric model. To run in forced mode, the model needs reasonably modern computer hardware (§1.4.1); in coupled mode, the hardware specifications will be primarily determined by the atmospheric model—unless an exceptionally simple atmospheric model is used—to which the 3D KPP model is coupled. In either case, installing and running the model requires that several pieces of freely available software also be installed on the system (§1.4.2). Potential users of the 3D KPP model should note that the model can currently be coupled only to one version of the OASIS coupler or to the GFS atmospheric model (§1.4.2). Users may—and are quite welcome to—write their own coupling routines to support the atmospheric model of their choice.

This documentation accompanies and describes the Fortran 77 version of the 3D KPP model. The KPP mixing scheme has been left almost entirely intact from that described by Large et al. (1994). Steve Woolnough and Nick Klingaman at the University of Reading wrote nearly all of the remaining code (i.e., the code outside of the mixing scheme itself) for the model. This version of the model can be run only on a single processor. Nick Klingaman is currently working on a Fortran 90 version of the model, with support for parallel processing as well as various code optimisations for speed and memory efficiency. There is no timetable for its release, but if you would like to be notified about its availability, please e-mail Nick.

You should have downloaded a distribution of the 3D KPP model along with these instructions. The distribution includes the source code for the model, as well as a reference case that you can run to ensure that you have compiled and installed the model properly on your system. These instructions will guide you through unpacking the model (§2.1), compiling the source code (§2.2) and running the reference case (§3). The reference case includes model output, to which you can compare the output produced by your local system.

Most of the options in the 3D KPP model are controlled via a set of namelist settings and corresponding

ancillary files. These options include the ability to linearly relax the model to user-specified sea-surface temperatures (SSTs), to specify heat corrections to constrain the mean state of the model, to blend model and user-specified SSTs and—when run in coupled mode—to restrict the coupling region to a user-specified box. These instructions contain details on all of the namelist options in the model (§4.1) as well as the configuration of all of the ancillary files that the model can use (§5); all ancillary files must be in the netCDF. There are also details concerning how to control the frequency of output from the model, as well as to control which fields the model outputs (§6). All output from the model is in netCDF. Further, these instructions also contain information on restarting an integration once it has completed, which may be useful on shared supercomputing facilities (§7).

If you encounter difficulties in compiling, configuring or running the model, you should in the first instance contact Nick Klingaman—via the e-mail address listed on the first page of these instructions—who has responsibility for routine maintenance of the code and this documentation. Please note, however, that we are not a technical support group and so may not be able to assist in all circumstances.

Good luck!

1.2 Credits, acknowledgments and references

Bill Large at NCAR wrote the original version of the K Profile Parameterization boundary-layer ocean model (Large et al., 1994). Steve Woolnough wrote the code that allows the model to run in three dimensions, as well as the first version of the OASIS coupling routines. He successfully coupled the model to the ECMWF monthly forecasting system and published a manuscript on the impact of refined vertical resolution and improved coupling frequency on MJO hindcasts (Woolnough et al., 2007). Nick Klingaman added many additional features to the 3D KPP model—the ability to relax to time-varying SSTs, pass-through of sea ice, depth-varying flux corrections, coupling weights—and modified the OASIS routines to work with the Hadley Centre's atmospheric model. Nick is also responsible for the GFS coupling routines. He conducted experiments with the Hadley Centre model on the impact of air–sea coupling for the intraseasonal variability of the Indian monsoon (Klingaman et al., 2009, and two other manuscripts in preparation).

If you publish research using the 3D KPP model, it would be appropriate to include Nick Klingaman as a (minor) author on the first paper that you publish. You should send a copy of your manuscript to Nick prior to submitting it to a journal. (In practice, it is likely that we will have been in touch to discuss your results long before your manuscript is ready to submit.) In subsequent papers, you should provide a statement in the "Acknowledgments" or similar section of your manuscript, stating that you used the 3D KPP model, written by Steve Woolnough and Nick Klingaman from code originally by Bill Large. We would appreciate receiving an electronic or paper copy of these manuscripts. You should cite Bill Large's original paper (Large et al., 1994) in any description of KPP.

1.3 Conventions

These instructions use the following conventions for formatting text:

- Important text is written in **bold face**.
- Important terms are written in *italics*.
- Technical terms are written in `monospace font` in the body of the text. These include names of files and directories, as well as compilation options and namelist settings.

- Commands for the user to enter are written in `monospace font` but are set in their own paragraphs. For example:

  ```
  cd scripts
  ```

  These commands should be entered exactly as written, with one exception: `[text in square brackets]` should be replaced as appropriate. The text in the brackets describes by what the text should be replaced. For example:

  ```
  cd [username]
  ```

  `[username]` should be replaced by the user's username on the local system.

1.4 System requirements

1.4.1 System specifications

The 3D KPP model has been used only on Solaris, Linux and Mac OS X. Other UNIX variants are likely to work as well. The model may run on other platforms (e.g., Windows) but these have not been tested.

Hardware requirements for the model vary based on the user's configuration, particularly in terms of the vertical resolution and the length of the run. When running with 60 vertical points, the model requires approximately 300 MB of RAM. In that configuration, an Intel 2.4 GHz. processor required approximately two minutes to complete one day of a forced simulation. This time will vary substantially based on the options that the user chooses to enable within the model.

1.4.2 Software packages

The following pieces of software are necessary to build and run the 3D KPP model in forced mode:

- The UNIX programs `tar` and `gzip`, which are distributed with most UNIX and Linux variants.
- The GNU `make` program, or another variant of `make` that is fully compatible with the GNU version.
- A Fortran 90 compiler. The model has been tested with `gfortran` version 4.3 on Linux and Mac OS X, as well as with `pathf90` (Pathscale) version 3 on Linux.
- The netCDF libraries, which are distributed by Unidata on their website `http://www.unidata.ucar.edu/software/netcdf`. The model has been tested with version 3.6.2 of these libraries.

Additionally, to run the test case you will need the `ksh` shell to process the setup script (§3.1). This is distributed with most UNIX and Linux variants. The script expects the shell to be at `/bin/ksh`, although this is easily reconfigurable if necessary.

To build the 3D KPP model in coupled mode using the OASIS coupler, you will also need the OASIS libraries. The model has been used successfully with version 2.4.1 of OASIS, which to my knowledge is no longer publicly distributed by its authors at CERFACS. The current incarnation of OASIS is version 3, but this is not guaranteed to work with the 3D KPP model; in fact, it will almost certainly not work. Information on how to link against the OASIS libraries can be found in §2.2.2.

To build the 3D KPP model coupled to the GFS, you will need the libraries that are built when the CFS is compiled. Information on how to link against these libraries can be found in §2.2.2.

To build the model using another coupling interface, you will need to modify the KPP coupling routines and link in any libraries that are required for your coupler of choice. This task is largely left up to you, but please contact me if you run into substantial difficulties.

1.4.3 Tested configurations

The 3D KPP model has been tested in forced mode with the following hardware and software configurations:

- An IBM Power 575 with an IBM Power 6 processor, AIX operating system, netCDF version 3.6.1, IBM XLF Fortran 90 compiler version 11.01
- A Cray XT4 with an AMD Opteron processor, Linux operating system, netCDF version 3.6.2, Pathscale Fortran 90 compiler version 3.00
- A Sun Fire V240 server with a SPARC processor, Solaris 10 operating system, netCDF version 3.6.2, Sun Studio Fortran 90 compiler version 12
- A MacBook with an Intel Core2 Duo processor, Mac OS X 10.5.6 operating system, netCDF version 3.6.3, GNU Fortran 90 (`gfortran`) compiler version 4.3.3
- A MacBook with an Intel Core2 Duo processor, Debian Linux operating system, netCDF version 3.6.2, GNU Fortran 90 (`gfortran`) compiler version 4.3.3

If you are successful in running the model on a different type of system from the one listed here, please let me know the details of the system and any difficulties you might have had in compiling and running the model. This will allow me to refine the code and this documentation.

2 Installing the Model

2.1 Unpacking the tarball

You should have received a file called `3D_kpp_r[?].tar.gz`, where `r[?]` refers to the *release number* of the code; the current release number is 1. This compressed tarball contains the model source code, along with sample ancillary files sufficient to make a five-day test run of the model. The file also includes the output from this run, so that you can compare your output to the example and thus ensure that your locally compiled version of the model is working properly.

To unpack the tarball, copy the file into the directory in which you intend to install the model and issue the following commands

```
gunzip 3D_kpp.tar.gz
tar -xvf 3D_kpp.tar
```

These commands should create a directory called `3D_kpp`, with the following sub-directories:

`ancillaries` - Contains the ancillary files required for the five-day test run of the model
`scripts` - Contains a script to set up the five-day test run
`src` - Contains the Fortran source code for the 3D KPP model
`test_output` - Contains the reference output for the five-day test run
`test_run` - The directory in which the test run will be conducted

2.2 Compiling the Model

2.2.1 Editing the `Makefile`

Before you can compile the model, you will need to edit the Makefile to correspond to your local environment. Enter the `src` directory. Examples of a Makefile are provided for the five configurations on which the model has been tested. These files are called `Makefile.linux_gfortran`, `Makefile.macosx_gfortran`, `Makefile.aix_xlf`, `Makefile.solaris_sunstudio`, and `Makefile.linux_pathscale`

You should edit the Makefile that corresponds most closely to the system you are using. If you cannot decide, use `Makefile.linux_gfortran`. I recommend saving your edited file to another name, something like `Makefile.[operating_system]_[compiler]`.

The only lines of the Makefile that you should need to edit are in the first block, following the comment "Make the 3D ocean model". Edit the lines as follows:

`F90` - The location of your Fortran 90 compiler

`CPP` - The location of your C Pre-processor, or simply `cpp` if directory containing `cpp` is in your `PATH`.

`F90_CPP_FLAG` - If your compiler requires a flag before a CPP declaration (i.e., a `-D` flag), then provide it here. This is an issue only for the IBM `xlf` compiler, which requires `-WF,` (note the comma).

`F90_FLAGS` - The compilation flags required to trigger the following options in your compiler:

- Processing by a C Pre-processor, like `cpp`. A C Preprocessor is necessary to ensure that the coupling routines are not included in the forced version of the model (see §1.4.2).
- A desired optimization level. `-O3` is recommended.
- Default double precision for all `REAL` variables. This is equivalent to specifying `KIND=8` for all `REAL` variables in the source code.
- Assumed fixed-format (i.e., Fortran 77 form of 72 characters per line) for all Fortran files.

Note that the IBM XLF compiler also requires the CPP option `-DOLD_NAME` to enable a piece of code in the 3D KPP model that enables the run-time environment to parse the "old-style" namelist that the model uses. This should be specified as a `F90_FLAG` as `${F90_CPP_FLAG}-DOLD_NAME`, where `${F90_CPP_FLAG}` is set appropriately as described above.

The following is a summary table of the options for each of the compilers with which the model has been tested:

Option	`gfortran`	`pathf90`	`f95` (Sun Studio)
C Preprocessor	`-x f77-cpp-input`	`-cpp`	`-fpp`
Optimization	`-O3`	`-O3`	`-O3`
Double-precision `REAL`s	`-fdefault-real-8`	`-r8`	`-xtypemap=real:64`
Fixed-format code	`-ffixed-line-length-72`	`-fixedform`	`-fixed`
Architecture-specific	None	None	None

Option	`xlf90` (IBM)
C Preprocessor	`-qsuffix=cpp=f`
Optimization	`-O3 -qstrict`
Double-precision `REAL`s	`-qrealsize=8 -q64`
Fixed-format code	`-qfixed`
Architecture-specific	`${F90_CPP_FLAG}-DOLD_NAME`

`OASIS_LIB` - If you plan to couple the model via OASIS, then specify the location of the OASIS `climmpi1.a` library here. Note that the model, as distributed, will work only with OASIS version 2 and only with the CLIM technique under MPI1 (see §1.4.2). If you do not plan to use OASIS, then leave this line unaltered.

`NCDF_LIB` - Edit the path following the `-L` option to reflect the directory on your system that contains the netCDF libraries. Leave the two options beginning with `-l` unaltered.

`NCDF_INC` - Edit the path following the `-I` option to reflect the directory on your system that contains the netCDF include files (i.e., the directory containing the file `netcdf.inc`).

`MPI_INC` - If you plan to couple the model via OASIS, then specify the directory on your system containing the MPI include files. Otherwise, leave this line unaltered.

`MPI_LIB` - If you plan to couple the model via OASIS, then specify the directory on your system containing the MPI libraries following the `-L` option; leave the two options beginning with `-l` unaltered. If you do not plan to use OASIS, then leave this line unaltered.

`CFS_LIB` - If you plan to couple the model to the GFS, then specify the directory on your system containing the CFS libraries following the `-L` option.

2.2.2 Executing the `Makefile`

Once you have finished editing a Makefile, you should copy it to the filename `Makefile`. For example, if I had used the `Makefile.linux_gfortran` makefile, I would run the command

```
cp Makefile.linux_gfortran Makefile
```

The `Makefile` contains three targets: one for the forced version of the model, one for the version coupled to OASIS and one for the version coupled to the GFS. You must request one of these targets when you compile the model. You cannot run the forced model in coupled mode, nor can you run the coupled model in forced mode.

To make an executable suitable for running the five-day test case distributed with this source code and documentation, you should compile the forced version of the model. To do so, issue the command

```
make forced
```

The compilation will then proceed. If you have correctly specified the compiler options in §2.2.1, then everything should go smoothly. If you encounter errors, try to determine which of the compiler options is at fault. In case of significant difficulties, contact me using the e-mail address shown on the first page.

To make an executable coupled via OASIS, issue the command

```
make oasis_coupled
```

Similarly, to make an executable coupled to the GFS, issue the command

```
make cfs_coupled
```

Following a successful compilation, you will have executable called `KPP_ocean`. Note that all three of these targets produce an executable with the same name. This means that if you intend to compile more than one target (i.e., you compile the forced version and then decide to try the OASIS coupled version), you must move or copy the executable from the first target before compiling the second; otherwise you will overwrite the first executable.

2.3 Cleaning up the `src` directory

To remove those pesky object (*.o) files, issue the command

```
make obj_clean
```

If you also wish to delete the `KPP_ocean` executable, issue the command

```
make clean
```

3 Running the five-day test case

The 3D KPP model comes with a test case and reference output, so that you can ensure your locally compiled copy of the model is working properly. The simulation is a five-day integration beginning on 1 May for the Indian Ocean and the West Pacific, on a regional grid extending from 30°S–30°N and 20°–180°E. The grid spacing is 1.25° in longitude and 0.83° in latitude. Sixty vertical points are used on a stretched grid, with a near-surface resolution of approximately 1 m.

This is a forced integration, driven by daily means of fluxes extracted from an AMIP-type run of the Hadley Centre Atmospheric Model (HadAM3). No advection is specified, but a May-mean heat correction is applied at all horizontal grid points and at all layers in the vertical. The temperatures in the bottom layer of the model (at 200 m) are fixed to the climatological May 204 m temperatures from the National Centre for Ocean Forecasting (NCOF) Forecast Ocean Assimilation Model (FOAM).

3.1 Setting up the run

To set up the five-day test case, enter the `scripts` directory and issue the command

```
./setup_test_run.ksh
```

This is a Korn shell script that copies the `KPP_ocean` executable into the `test_run` directory. It also creates symbolic links from the ancillary files in the `ancillaries` directory to the necessary locations in the `test_run` directory.

The `setup_test_run.ksh` script assumes that the Korn shell exists at `/bin/ksh`. If the shell exists at a different location, then you will need to edit the first line of the script to reflect the shell's location on your system. If you do not have the Korn shell and do not wish to install it, you may need to re-write the script to reflect the syntax of the shell of your choice.

3.2 Execute the run

To run the five-day test case, enter the `test_run` directory and issue the command

```
./KPP_ocean > ./test_run.out &
```

This will run the executable you built in §2.2.2. The file `3D_ocn.nml` is a namelist file that contains various settings related to the model grid, the ancillary files and options such as heat corrections and climatologies. You can find out more about the namelist and the meaning of the various options in §4.1. Note that `3D_ocn.nml` is not required as standard input; the model opens the file on unit 75 during execution.

If the executable has been compiled correctly, the model should now run the test case. Output will be directed to the file `test_run.out`. You can keep track of the model integration by issuing the command

```
tail -f test_run.out
```

As this test case uses 60 vertical levels, it will require some time to complete the integration. An Intel Core2 Duo 2.4 GHz. processor required approximately 8 minutes to finish the run.

3.3 Viewing output

When the test case completes, it will leave a netCDF output file called `KPPocean_0005_means.nc` (see §6 for further details on output files). This file contains daily-mean output of the following variables

netCDF name	Description
T	Temperature at all vertical points
rho	Density (ρ) at all vertical points
cp	Specific heat (C_p) at all vertical points
hmix	Mixing depth
cplwght	Coupling weight

You can view this output using any software package capable of viewing standard netCDF files. If you do not have such a package, I highly recommend `ncview`, which is available to download from `http://meteora.ucsd.edu/~pierce/ncview_home_page.html`.

3.4 Comparing against the reference output

The directory `test_output` contains the reference output for the five-day test case. This file is also called `KPPocean_0005_means.nc`. You should compare your output (in the `test_run` directory) against the reference output to ensure that the simulation ran correctly. You can make this comparison in any graphics package that can read netCDF files, but it is easiest to use the `NCO` (NetCDF Operators) software package. If you do not already have this package, you can download it from `http://nco.sourceforge.net`.

Using the `ncdiff` program contained in the `NCO` package, you can enter the `test_run` directory and issue

```
ncdiff KPPocean_0005_means.nc ../test_output/KPPocean_0005_means.nc
diff_from_reference.nc
```

The file `diff_from_reference.nc` now contains the difference between your output and the reference output, calculated as your output minus the reference. You can then view this netCDF file in your graphics package of choice.

Some small differences between your output and the reference output can be expected, particularly if you have used a different compiler, hardware architecture and operating system than the one used to generate the reference output: a MacBook running Debian Linux on an Intel Core2 Duo, using the `gfortran` 4.3.3 compiler. These differences may be as much as $\pm 0.1^{\circ}$C in temperature, although this is an extreme value and should occur only at a very small number of gridpoints.

Once you are convinced that your test integration matches the reference solution, you should proceed to configuring the model for your own purposes. The following sections provide some guidance on this.

4 Configuring the Model

4.1 Namelist options

This subsection describes the options that can be set in the `3D_ocn.nml` namelist file. The file is broken into namelist groups, with each group containing a set of related options. The following list of options is organized by these namelist groups. Where applicable, the default value for each option is given. If you do not specify a value in `3D_ocn.nml`, the model uses the default value.

Namelist group `NAME_CONSTANTS`

Namelist option	Defaut value	Description
`grav`	9.816 m s^{-2}	Acceleration due to gravity
`vonk`	0.4	Von Karman's constant
`sbc`	5.67x10^{-8} W m^{-2} K^{-4}	Stefan-Boltsmann constant
`twopi`	8 $\left[\tan^{-1}(1.0)\right]$	Twice the value of π to machine precision
`onepi`	$\frac{\texttt{twopi}}{2.0}$	The value of π to machine precision
`TK0`	273.15 K	The value of 0°C in Kelvin
`spd`	86400.0 s	The number of seconds per day
`dpy`	360.0 days	The number of days per year (not currently used in the model)
`epsw`	1.0	A correction factor for the departure of water from a black body
`albocn`	0.06	The albedo of sea water
`EL`	2.5x10^{6} J kg^{-1}	The latent heat of evaporation for water
`SL`	2.5122x10^{6} J kg^{-1}	The latent heat of evaporation for ice
`FL`	3.34x10^{5} J kg^{-1}	The latent heat of fusion for ice
`FLSN`	3.34x10^{5} J kg^{-1}	The latent heat of fusion for snow

Namelist group `NAME_PROCSWIT`

Namelist option	Defult value	Description
`LKPP`	`.TRUE.`	Enables KPP boundary-layer mixing scheme
`LRI`	`.TRUE.`	Enables Richardson-type mixing scheme in the "interior" of the boundary layer, below the model-diagnosed mixing depth
`LDD`	`.FALSE.`	Enables double-diffusion calculations (greatly increases run time)
`LICE`	`.FALSE.`	Enables simple sea-ice submodel (not implemented in this version)
`LBIO`	`.FALSE.`	Enables simple ocean biology submodel (not implemented)
`LNBFLX`	`.FALSE.`	Enables a no-flux boundary condition (no diffusion or viscosity) for the bottom of the vertical domain
`L_SSref`	`.TRUE.`	Enables calculating the surface salinity as an offset from a reference surface salinity. The reference surface salinity is taken as the initial condition.

Namelist group NAME_DOMAIN

Namelist option	Default value	Description
dmax	0 m	The maximum depth for the ocean model (i.e., the bottom of the vertical domain); takes a **positive** value.
L_STRETCHGRID	.FALSE.	Specifies whether the vertical grid is stretched (.TRUE.) or regular (.FALSE.). If set to .TRUE., you must also set dscale.
dscale	0.0	A scaling factor to use in stretching the vertical grid. For the actual equation used, see §4.2.2. Positive values result in more vertical levels being placed near the ocean surface.
alat	0.0	The latitude of the bottom-left corner of the horizontal domain; sign convention is **positive north**.
alon	0.0	The longitude of the bottom-left corner of the horizontal domain; sign convection is **positive east**.
L_REGGRID	.TRUE.	Specifies whether the horizontal grid is regular in latitude and longitude. If set to .FALSE., the model ignores the delta_lat and delta_lon options below; instead, the model reads the grid in from the land-sea mask ancillary file.
delta_lat	2.5	The spacing in latitude between grid points, assuming a regular grid. If your grid is irregular, you must set the L_REGGRID option to .FALSE. to read the grid in from the land-sea mask file.
delta_lon	3.75	The spacing in longitude between grid points, assuming a regular grid. If your grid is irregular, you must set the L_REGGRID option to .FALSE. to read the grid in from the land-sea mask file.

Namelist group NAME_START

Namelist option	Default value	Description
L_INITDATA	.TRUE.	Indicates whether initial conditions will be specified for the model. Currently, you **must** specify an initial condition if you are not conducting a restart run. Leave this option set to .TRUE..
initdata_file	None	The location of the netCDF file containing the initial conditions for the model; see §5.1 for the variables required to initialize KPP.
L_INTERPINIT	.TRUE.	Indicates whether the intial conditions need to be interpolated onto the KPP vertical grid. Note that the model is incapable to performing any horizontal interpolation; see §5.1 for further details.
L_RESTART	.FALSE.	Indicates whether the current integration should resume from the output of a previous integration; see §7 for further information.
restart_infile	fort.30	The location of the file containing the fields necessary to restart the model from a previous integration. This file must be in the internal file format required by the model; see §7 for further information.

Namelist group NAME_TIMES

Namelist option	Default value	Description
dtsec	None	The time (in seconds) between updates to the surface forcing data. If KPP is running in forced mode, this is frequency at which the surface forcing is updated from the forcing file specified in forcing_file in the NAME_FORCING group. If KPP is running in coupled mode, this is the coupling frequency.
startt	None	The time (in days) at which the integration begins. The time dimensions of some of the ancillary files must match KPP's own reckoning of the time; see §5 for further information. If this is a restart run, this time must match the time in the restart file; see §7.
finalt	None	The time (in days) at which the integration ends. This time will be written to the restart file, if one is requested. Upon restarting, this final time must be specified as the starting time (startt); see §7. The total length of the run must be an integer multiple of the model timestep, or else the run will fail and an error message will be generated. The timestep is specified via ndtocn below.
ndtocn	1	The number of model timesteps per update to the surface forcing data. This means that the timestep of KPP (in seconds) is dtsec divided by ndtocn. This is the only way to specify the model's timestep.

Namelist group NAME_COUPLE – **Note** that you may wish to enable several of these options—particularly the ones regarding reference SSTs—even if you are running the model in forced mode.

Namelist option	Default value	Description
L_COUPLE	Varies	Specifies whether the model is to be run in coupled (.TRUE.) or forced (.FALSE.). If the model has been compiled in forced mode, the default value is .FALSE.; if the model has been compiled in coupled mode, the default value is .TRUE.
ifirst	None	The x (longitude) position of the first coupled point on the grid of the atmospheric model to which the 3D KPP model is coupled. Note that this is the **position** of the point, **not** the longitude value in degrees. Also note that if you are specifying reference SSTs via L_CLIMSST, the domain of the atmospheric model must match the domain of the reference SSTs. In almost all cases, this domain will be the global atmospheric model domain. Together with ilast, jfirst and jlast, this specifies the domain within which the 3D KPP model is coupled to the atmospheric model.
ilast	None	The x (longitude) position of the last coupled point on the grid of the atmospheric model to which the 3D KPP model is coupled. See ifirst for further details.
jfirst	None	The y (latitude) position of the first coupled point on the grid of the atmospheric model to which the 3D KPP model is coupled. See ifirst for further details.

jlast	None	The y (latitude) position of the last coupled point on the grid of the atmospheric model to which the 3D KPP model is coupled. See ifirst for further details.
L_CPLWGHT	.FALSE.	Specifies whether to use an ancillary netCDF file of coupling weights. The coupling weight is used to define a blending of model and reference (i.e., climatological or observed) SSTs. If this is set to .FALSE., then the KPPs SSTs are returned to the atmospheric model at every grid point; if set to .TRUE., you must specify an ancillary file via cplwght_file, as well as reference SSTs via L_CLIMSST and sstin_file. Coupling weights apply **only** to the SSTs returned to the atmosphere, not to the SSTs written to KPP's own output files.
cplwght_file	None	The location of the ancillary netCDF file giving the coupling weights. For more information on coupling weights, see §5.2.
L_OUTKELVIN	.FALSE.	Specifies whether to return the SST to the atmospheric model in Kelvin (.TRUE.) or Celsius (.TRUE.). This option is enabled only when the model is coupled, whether to the GFS or via OASIS.
L_UPDCLIM	.FALSE.	Specifies whether the climatologies of SST, sea ice and heat corrections are to updated (i.e., whether the climatologies are time-varying). Note that this does not **enable** the use of any of these climatologies; for that, see the L_CLIMSST, L_CLIMICE and L_FCORR options.
L_CLIMSST	.FALSE.	Indicates whether to read in a climatological reference SST. Reference SSTs are commonly used for two purposes: to linearly relax the model SSTs (see the L_RELAX option) and, when coupled, to provide a global SST field to an atmospheric model. In the second case, the KPP SSTs are combined with the reference SSTs; this combination is controlled via the coupling weights (see L_CPLWGHT).
sstin_file	None	The location of a netCDF ancillary file containing reference SSTs. For more information on the SST ancillary file, see §5.5.
ndtupdsst	None	The number of ocean model timesteps between updates to the reference SSTs; this option takes effect only if L_UPDCLIM is set to .TRUE.. The time dimension of the ancillary file must contain a value that matches the midpoint of each update interval. For example, if ndtupdsst is set to 24 and the model has a timestep of one hour, then the SST ancillary file must contain a time dimension with values (in days) 0.5, 1.5, 2.5 and so on.
L_CLIMICE	.FALSE.	Indicates whether to read in a climatological field of sea-ice concentrations. These are necessary only if the atmospheric model to which KPP is coupled expects to receive a sea-ice field via the coupler. The model does not modify the sea-ice values; it merely passes them to the atmospheric model via the coupler.
icein_file	None	The location of a netCDF ancillary file containing climatological sea-ice concentrations. For more information on the sea ice ancillary file, see §5.6.

ndtupdice	None	The number of ocean model timesteps between updates to the sea-ice concentrations; this option takes effect only if L_UPDCLIM is set to .TRUE.. The time dimension of the ancillary file must contain a value that matches the midpoint of each update interval.

Namelist group NAME_ADVEC

Namelist option	Default value	Description
L_ADVECT	.FALSE.	Specifies whether to use user-prescribed horizontal advection in the model. Advection terms are specified via the advect_file ancillary file.
advect_file	None	The location of the ancillary netCDF file giving the prescribed horizontal advection terms. For more information on horizontal advection, see §5.4.
L_RELAX	.FALSE.	Specifies whether to linearly relax the KPP SSTs towards user-specified SSTs. The SSTs must be specified via the sstin_file option in the L_COUPLE namelist. The relaxation is calculated so as to correct the entire model-diagnosed mixed-layer depth by the bias in the SSTs.
relax_in	.FALSE.	Gives the linear relaxation timescale in days, with one value per latitude band. These values can be given using standard Fortran namelist syntax (i.e., nlat*timescale). Thus, to specify that the first five latitude bands have a relaxation timescale of 20 days and the next 10 have a relaxation timescale of 30 days, the statement would be relax_in=5*20,10*30. For no relaxation in a particular latitude bands, use a value of zero (0). Note that you must specify a value for all latitude bands. Further, note that it is not currently possible to vary the relaxation timescale within a single latitude band.
L_RELAX_CALCONLY	.FALSE.	If set to .TRUE., the relaxation term is calculated but never applied to the model SSTs. Note that this option will not give the same values as L_RELAX, because the relaxation terms will always be calculated for completely unconstrained SSTs.

Namelist group NAME_PARAS

Namelist option	Default value	Description
L_JERLOV	.TRUE.	Specifies whether to use user-provided Jerlov water types. If set to .FALSE., all gridpoints are set to a Jerlov water type of IB.
paras_file	3D_ocnparas.nc	The location of the ancillary netCDF file giving the Jerlov water type at each KPP horizontal gridpoint. For more information on Jerlov water types, see §5.10.

Namelist group NAME_OUTPUT

Namelist option	Default value	Description
L_OUTPUT_INST	.TRUE.	Specifies whether to output instantaneous fields from the model to a netCDF file. Fields are output every ndtout timesteps. The resulting files are called KPPocean_[last_output_time].nc; a new file is created every five days of the integration. For further information on instantaneous output files, see §6.3.
L_OUTPUT_MEAN	.FALSE.	Specifies whether to output mean fields from the model to a netCDF file. The meaning period is defined by ndtout_mean, with the value in units of timesteps. The meaning period is the same as the output frequency; it is not possible to, for example, output a six-hourly mean once per day. The resulting fields are called KPPocean_[last_output_time]_means.nc; a new file is created every five days of the integration.
L_VAROUT	16*.TRUE.	Specifies which of the available model three-dimensional (lon x lat x depth) fields to output to the instantaneous output files. This namelist option accepts a list of logical flags (.T. or .F.) separated by commas; the flags correspond to the list of available three-dimensional fields in §6.1.
L_MEAN_VAROUT	16*.FALSE.	Specifies which of the available model three-dimensional (lon x lat x depth) fields to output to the mean output fields. This option has no effect unless L_OUTPUT_MEAN is set to .TRUE.. This namelist option accepts a list of logical flags (.T. or .F. separated by commas; the flags correspond to the list of available three-dimensional fields in §6.1.
L_SINGOUT	5*.TRUE.	Specifies which of the available single-level (lon x lat) fields to output to the instantaneous output files. This namelist option accepts a list of logical flags (.T. or .F.) separated by commas; the flags correspond to the list of available single-level fields in §6.2.
L_MEAN_SINGOUT	5*.FALSE.	Specifies which of the available single-level (lon x lat) fields to output to the mean output files. This option has no effect unless L_OUTPUT_MEAN is set to .TRUE.. This namelist option accepts a list of logical flags (.T. or .F.) separated by commas; the flags to the list of available single-level fields in §6.2.
ndtout	1	The frequency in units of model timesteps (see ndtocn) at which to output the instantaneous fields specified in L_VAROUT and L_SINGOUT.
ndtout_mean	24	The frequency in units of model timesteps (see ndtocn) at which to output the mean fields specified in L_VAROUT_MEAN and L_SINGOUT_MEAN. Note that this option also sets the meaning period; it is not possible to distinguish between the meaning period and the output frequency.

ndt_per_file	Varies	The frequency in units of model timesteps (see ndtocn at which to create a new output file. This controls the amount of output written to any single file. The default is five days $\left(\frac{\texttt{5 x dtsec x ndt\_per\_file}}{\texttt{ndtocn x spd}}\right)$ when run in forced mode or when coupled to OASIS, or one day when coupled to the GFS.
L_RESTARTW	.TRUE.	Specifies whether to write an output file at the end of the integration. By default, the file is written to the file fort.31. This can be altered via the restart_out_file option below.
restart_out_file	fort.31	The location to which to write the restart file containing the fields necessary to restart the model from the end of the current integration. This file is in an internal format; see §7 for further information.

Namelist group NAME_FORCING

Namelist option	Default value	Description
L_FLUXDATA	.FALSE.	Specifies whether to use a user-provided ancillary file of surface forcing. This is used only if L_COUPLE is set to .FALSE.. If the model is not coupled and no forcing file is provided, a simple, time-invariant forcing is applied at all horizontal grid points.
forcing_file	1D_ocean_forcing.nc	The location of the ancillary netCDF file containing the surface forcing data to be used in the forced version of the model. The file must contain the following fields: net shortwave radiation, net longwave radiation, sensible heat flux, latent heat flux, zonal and meridional surface wind stress and precipitation minus evaporation. For further information about this file, see §5.9.
L_FCORR	.FALSE.	Specifies whether to use a user-provided ancillary file of surface-only heat corrections. The heat corrections must be in units of W m^{-2}. The corrections are intended to correct the entire model-diagnosed mixed-layer depth by the bias at the surface. The model can calculate and output these calculations using the L_RELAX option with a specified relaxation timescale relax_in. For further information on heat corrections, see §5.8.
L_FCORR_WITHZ	.FALSE.	Specifies whether to use a user-provided ancillary file of depth-varying heat corrections. These heat corrections must be in units W m^{-3}. The corrections are intended to correct each layer in the vertical independently. Thus, the ancillary file must be at the same vertical resolution as the model. The model cannot calculate these corrections; they must be calculated offline. For more information on heat corrections, see §5.8.
fcorrin_file	None	The location of the ancillary netCDF file containing the heat corrections, either at the surface only (L_FCORR) or as a function of depth (L_FCORR_WITHZ). For details on the configuration of this file, see §5.8.

ndtupdfcorr	None	The frequency (in timesteps, see ndtocn) with which to update the heat corrections. The ancillary netCDF file specified in fcorrin_file must contain a time dimension with one value (in days) for each update time.
L_VARY_BOTTOM_TEMP	.FALSE.	Specifies whether to replace the KPP bottom-layer temperature with a user-specified value. If set to .FALSE. then the treatment of the bottom layer is given by LNBFLX in the NAME_PROCSWIT namelist. If set to .TRUE. then the user must provide an ancillary netCDF file containing the temperatures to be used at each gridpoint; see bottomin_file below.
bottomin_file	None	The location of the ancillary netCDF file containing the temperatures for the bottom layer of the model. These temperatures will overwrite the model's temperature at every timestep, regardless of the setting of LNBFLX. For more information on the configuration of this file, see §5.7.
ndtupdbottom	None	The frequency (in timesteps, see ndtocn) with which to update the bottom-layer temperatures from the ancillary netCDF file. Note that the bottom-layer temperatures in the model are replaced every timestep regardless of this setting; this setting controls how frequently the value used for the bottom-layer temperature is updated from the netCDF file. The ancillary netCDF file specified in bottomin_file must contain a time dimension with one value (in days) for each update time.

Namelist group NAME_LANDSEA

Namelist option	Default value	Description
L_LANDSEA	.FALSE.	Specifies whether to use a user-provided ancillary file for the land/sea mask and the depth of the ocean. If set to .FALSE., the model assumes that all points in the domain are ocean points and that the ocean depth is equal to the vertical domain of the model.
landsea_file	None	The location of the ancillary netCDF file containing the land/sea mask and the depth of the ocean. For more information on this netCDF file, see §5.3.

4.2 Model parameters

To configure KPP to run in any three-dimensional domain other than the default domain provided with the five-day test case—the Indian Ocean and the West Pacific—you will need to edit the file parameter.inc. The following sub-sections describe how to alter the horizontal domain (§4.2.1) and the vertical resolution (§4.2.2) of the 3D KPP model.

Please note that you will need to **recompile** the model after making any change to the parameter.inc file. You can find details on compiling the model in §2.2.2.

4.2.1 Horizontal domain

The 3D KPP model needs to know how many one-dimensional water columns to configure in the latitudinal and longitudinal direction. If you are running the model in forced mode, this is the same as the horizontal dimensions of your forcing ancillary file (§5.9). If you are running the model in coupled mode, this is the same as the horizontal dimensions of your coupling region.

Once you know the dimensions of your domain, edit the `parameter.inc` file and change the `NX` and `NY` variables to the number of points in longitude and latitude, respectively.

If you are running the model in coupled mode, you will also need to provide the horizontal dimensions of the atmospheric model to which you are coupling. Edit the `parameter.inc` file and change the `NX_GLOBE` and `NY_GLOBE` variables to the number of points in longitude and latitude, respectively.

Note that the `parameter.inc` does not specify the horizontal *resolution* of the grid, only the number of points. The horizontal resolution is specified via the `delta_lat` and `delta_lon` namelist variables for a regular grid, or via a land/sea mask ancillary file (§5.3) for a non-regular grid; see the description of the `NAME_DOMAIN` namelist options in §4.1 for further information.

4.2.2 Vertical resolution

To alter the number of vertical points in each one-dimensional water column, you must change the `NZ` variable in the `parameter.inc` file. `NZ` gives the number of "layers" in the model, not including the resting bottom layer. Thus, specifying `NZ=59` will result in 60 vertical points in the output netCDF files, including the bottom layer.

You should not need to alter any of the other vertical-grid parameters in the `parameter.inc` file, as they all depend upon the value of `NZ`.

Typically, KPP operates on a stretched vertical grid, with more points near the surface of the ocean. This near-surface "stacking" of the points is controlled by the `dscale` namelist option. The equation used to calculate the position of each vertical point is as follows (taken from the `INIT_ENV` subroutine in `init.f`):

```
DO i=1,NZ
```

$$hm(i) = \frac{\texttt{DMAX} * \left(1.0 - e^{-\texttt{dscale}}\right)}{\texttt{NZ} * \texttt{dscale} * \left(1.0 - \left(\frac{i-0.5}{\texttt{NZ}}\right) * \left(1.0 - e^{-\texttt{dscale}}\right)\right)} \tag{1}$$

```
ENDDO
```

Note that `DMAX` is the depth of the column; this can be altered in the namelist (see §4.1).

5 Ancillary files

This section describes the required configuration of the ancillary files for the 3D KPP model. All of these files are optional **except** for the initial conditions file, which is required for non-restart runs; their use can be controlled by the relevant namelist options given in §4.1. All ancillary files **must** be in netCDF format, as the model is currently incapable of reading any other type of file. The following subsections give the netCDF dimension and variable names and sign conventions that the model expects. Most of the subroutines to read these files are contained in the file `ncdf_in.f`.

Note that an example of each file is provided in the `ancillaries` directory of the distribution.

5.1 Initial conditions

The initial conditions ancillary file is **required** if the run is not a restart run (i.e., if `L_RESTART=.F.`; see §4.1 and §7). The model will attempt to read the initial conditions from the file specified in the `initdata_file` namelist option.

An example of this ancillary file is provided as `may1_initial_conditions.nc` in the `ancillaries` file.

The netCDF file must have the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`
zvel	Depth coordinate for velocity	User-specified
zsal	Depth coordinate for salinity	User-specified
ztemp	Depth coordinate for temperature	User-specified

Note that the model will interpolate from the provided depth coordinates to the vertical grid of the model. Thus, the initial conditions do not need to be provided on the model vertical grid. There is, however, no facility to interpolate horizontally; the initial conditions must have the same horizontal grid as the 3D KPP model. The horizontal grid of the model is determined by the settings in the `parameter.inc` file; see §4.2.1 for further information.

The netCDF file must have the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
zvel(zvel)	Depth coordinate for velocity	m	Negative below surface
zsal(zsal)	Depth coordinate for salinity	m	Negative below surface
ztemp(ztemp)	Depth coordinate for temperature	m	Negative below surface
u(longitude,latitude,zvel)	Zonal velocity	m s^{-1}	Positive westerly
v(longitude,latitude,zvel)	Meridional velocity	m s^{-1}	Positive southerly
temp(longitude,latitude,ztemp)	Temperature	°C	Always positive
sal(longitude,latitude,zsal)	Salinity	$^0/_{00}$	Always positive

5.2 Coupling weights

An ancillary file of coupling weights is **required** if the `L_CPLWGHT` namelist option is set to `.TRUE.`. The model will attempt to read the coupling weights from the file specified in the `cplwght_file` namelist option. Coupling weights are useful only if the user also provides prescribed SSTs via the `L_CLIMSST` and `sstin_file` options; see §4.1 and §5.5 for further information,

Coupling weights are used to combine model and prescribed SSTs when KPP is coupled to an atmospheric model. The coupling weight itself must be a value between 0 and 1, specifying the weight to give to the **model** SSTs. A weight of 1 minus the coupling weight is given to the prescribed SSTs provided in the `sstin_file` ancillary file. The equation is therefore

$$SST_{atmos} = SST_{kpp} * \alpha + (1 - \alpha) * SST_{user} \quad (2)$$

where SST_{kpp} is the model SST, SST_{user} is the user-prescribed SST (e.g., climatological, observed), α is the coupling weight and SST_{atmos} is the SST returned to the atmospheric model.

Note that coupling weights **do not** affect the SST in KPP itself, only the SST returned to the atmospheric model. Thus, coupling weights cannot be used as a means to relax the model SSTs to some specified value; to accomplish that, you must use the `L_RELAX` and `relax_in` namelist options (§4.1) or an ancillary file of heat corrections (§5.8).

An example of this ancillary file is provided as `coupling_weight_5ptblend.nc` in the `ancillaries` directory.

The netCDF file must have the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`

The netCDF file must have the following variables in the following dimensions:

netCDF name	Description	Units	Sign convention
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
alpha(longitude,latitude)	Coupling weight	None	Always positive

5.3 Land/sea mask and ocean depth

An ancillary file giving the land/sea mask and ocean depth is **required** if `L_LANDSEA` is set to `.TRUE.` or if `L_REGGRID` is set to `.FALSE.`. In the latter case, this ancillary file is used to provide the horizontal coordinates (longitude and latitude) to the model when an irregular horizontal grid is used.

The land/sea mask should be a variable of type `float`: a value of less than 0.5 indicates an ocean point; a value of 0.5 or greater indicates a land point. KPP ignores all land points in the domain. In the output files, these points are marked with a missing value.

The ocean depth is used to prevent KPP from mixing below the physical bottom of the ocean. Values specified in this variable will constrain the mixing depth of KPP to be between the surface and the specified values. This is important in regions such as the Maritime Continent, where the ocean in some places is less than 100 metres deep. Note that the values specified here do not constrain the vertical domain of the model, only the calculated mixing depth. If you do not wish to constrain the mixing depth at all, then set this variable to some large negative number.

An example of this ancillary file is provided as `land-sea_mask_ocean_depth.nc`.

The netCDF file must have the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`

The netCDF file must have the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
lsm(longitude,latitude)	Land/sea mask	None	Always positive
max_depth(longitude,latitude)	Depth of ocean	None	Negative below surface

5.4 Horizontal advection

Bill Large's original code provides for several different advection options for temperature and salinity. Advection may be enabled via the `L_ADVECT` namelist option; if this is set to `.TRUE.` then an ancillary file of advection options and terms is `required` and must be specified via the `advect_file` namelist option (§4.1).

At each horizontal gridpoint, the user may choose one or more of the following advection modes for temperature or salinity or both. Note that the number of the mode here corresponds to the integer that must be specified in the `nmode_tadv` (for temperature) or `nmode_sadv` (for salinity) netCDF variables. Further information on these advection options may be found in the `ocn.f` file, particularly in the `rhsmod` subroutine.

Mode number	Description
1	Steady upper-layer (i.e., surface) horizontal advection
2	Steady horizontal advection within the model-diagnosed mixed layer
3	Steady horizontal advection throughout the water column (to `dmax`)
4	Steady vertical advection from 100 m to the bottom of the domain
5	Steady diffusion at the bottom vertical point
6	Seasonally varying mixed-layer horizontal advection
7	Seasonally varying thermocline horizontal advection

Please note that these advection options have not been rigorously tested at Reading; the user may need to modify the code in `ocn.f` to allow these options to work correctly, particularly for options 6 and 7.

The ancillary file should contain three variables for temperature and three variables for salinity. One variable specifies the **number** of modes the user is selecting above; the second variable specifies the **mode number** from the table above; the third variable specifies the **advection term** itself.

The ancillary file should also contain a dimension called `mode`. This dimension allows the user to specify more than one advection mode. The length of this dimension should be equal to the value of `maxmodeadv` in `parameter.inc`. This parameter specifies the maximum number of advection modes that can be requested at any grid point. The default is `maxmodeadv=6`. Reducing this value may slightly decrease the amount of memory that the model requires.

An example of this ancillary file is provided as `default_advection.nc`.

The netCDF file must have the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`
mode	Mode coordinate	Equal to `maxmodeadv` in `parameter.inc`

The netCDF file must have the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
mode(mode)	Mode coordinate	None	Always positive
nmode_tadv(longitude,latitude)	Number of modes for T	None	Always positive
nmode_sadv(longitude,latitude)	Number of modes for S	None	Always positive
mode_tadv(longitude,latitude,mode)	Mode(s) to use for T	None	Always positive
mode_sadv(longitude,latitude,mode)	Mode(s) to use for S	None	Always positive
tadv(longitude,latitude,mode)	Advection term(s) for T	W m^{-2}	Positive to warm ocean
sadv(longitude,latitude,mode)	Advection term(s) for S	mm s^{-1}	Positive to freshen ocean

5.5 SSTs

An ancillary file of SSTs is **required** if the `L_CLIMSST` namelist option is set to `.TRUE.`. The model will attempt to read the SSTs from the netCDF file specified in the `sstin_file` namelist option (§4.1). If the `L_UPDCLIM` option is set to `.TRUE.`, then this ancillary file must contain **time-varying** SSTs, with one value (or horizontal field of values) for each update time; the update frequency is controlled by the `ndtupdsst` namelist option, in units of model timesteps.

For time-varying SSTs, the `time` variable described below must contain a value for each update time, in units of days. The value should be specified as the midpoint of the update interval. For example, consider the following namelist settings:

```
startt=0000.0
dtsec=86400.
ndtocn=24
ndtupdsst=24
```

This specifies a model timestep of one hour (86400 seconds divided by 24) and an update frequency of 24 timesteps, or one day. The model begins its integration at day 0. The first value of the time variable in the SST ancillary file must be 0.5; the second must be 1.5; the third 2.5; and so on. Note that if the value of `startt` were non-zero, the values of the time variable would need to be adjusted accordingly. Thus, the times in the netCDF file must correspond to the model's internal time.

Note that if the model is being coupled to an atmospheric model (`L_COUPLE=.TRUE.`), then this ancillary file must contain **global** SSTs. When the model is run in forced mode, the SSTs may be either global or restricted to the region in which KPP is running.

Also note that SSTs may be specified in either Kelvin or degrees Celsius. If the SSTs are provided in Kelvin, the model will convert them to Celsius for its own use.

An example of a time-varying SST ancillary file is provided as `global_ssts_daily_clim.nc` in the `ancillaries` directory.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc` if forced or `NX_GLOBE` if coupled
latitude	Latitude	Equal to `NY` in `parameter.inc` if forced or `NY_GLOBE` if coupled
time	Time (only if `L_UPDCLIM`)	Equal to $\frac{\texttt{ndtocn}*86400*(\texttt{finalt}-\texttt{startt}+1)}{\texttt{dtsec}*\texttt{ndtupdsst}}$

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
time(time)	Time (only if `L_UPDCLIM`)	Days	Always positive
sst(longitude,latitude,time)	Sea-surface temperature	Kelvin or °C	Always positive

5.6 Sea ice

An ancillary file of sea ice concentrations is **required** if the `L_CLIMICE` namelist option is set to `.TRUE.`. The model will attempt to read the sea ice concentrations from the netCDF file is specified in the `icein_file` namelist option (§4.1). If the `L_UPDCLIM` option is set to `.TRUE.`, then this ancillary file must contain **time-varying** sea ice concentrations, with one value (or horizontal field of values) for each update time; the update frequency is controlled by the `ndtupdice` namelist, in units of model timesteps.

For a description of how to configure the `time` variable for time-varying sea ice concentrations, see §5.5 and the example contained therein.

Sea ice concentrations are required only if the 3D KPP model is coupled to an atmospheric model that requires the ocean model to pass ice concentrations. The 3D KPP model does not modify or even use the sea ice concentrations. Otherwise, this option should not need to be used.

An example of a time-varying sea ice ancillary file is provided as `global_ice_daily_clim.nc` in the `ancillaries` directory.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`
time	Time (only if `L_UPDCLIM`)	Equal to $\frac{\texttt{ndtocn}*86400*(\texttt{finalt}-\texttt{startt}+1)}{\texttt{dtsec}*\texttt{ndtupdice}}$

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
time(time)	Time (only if `L_UPDCLIM`)	Days	Always positive
iceconc(longitude,latitude,time)	Sea ice concentrations	Any	Any

5.7 Bottom-layer temperatures

An ancillary file of bottom-layer temperatures is **required** if the L_VARY_BOTTOM_TEMP namelist option is set to .TRUE.. The model will attempt to read the bottom-layer temperatures from the netCDF file specified in the bottomin_file namelist option (§4.1). If the L_UPDCLIM option is set to .TRUE., then this ancillary file must contain **time-varying** bottom-layer temperatures, with one value (or a horizontal field of values) per update time; the update frequency is controlled by the ndtupdbottom option, in units of model timesteps.

For a description of how to configure the time variable for time-varying bottom-layer temperatures, see §5.5 and the example contained therein.

These user-specified bottom-layer temperatures will overwrite the KPP bottom-layer temperature at every model timestep. This option thus provides a means to constrain the bottom layer of the model to a fixed or time-varying climatology. Note that there is also a namelist option LNBFLX which specifies a no-flux boundary condition for the bottom model layer. This will constrain the temperature to the initial condition, rather than to a separate user-specified value.

An example of a time-varying bottom-layer temperatures ancillary file is provided as bottomtemp_daily_clim.nc.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to NX in parameter.inc
latitude	Latitude	Equal to NY in parameter.inc
time	Time (only if L_UPDCLIM)	Equal to $\frac{\texttt{ndtocn}*86400*(\texttt{finalt}-\texttt{startt}+1)}{\texttt{dtsec}*\texttt{ndtupdbottom}}$

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
time(time)	Time (only if L_UPDCLIM)	Days	Always positive
T(longitude,latitude,time)	Bottom-layer temperatures	Kelvin or °C	Always positive

5.8 Heat corrections

An ancillary file of heat corrections is **required** if either the L_FCORR or L_FCORR_WITHZ namelist options are set to .TRUE.. The model will attempt to read the heat corrections from the netCDF file specified in the fcorrin_file namelist option (§4.1). If the L_UPDCLIM option is set to .TRUE., then this ancillary file must contain **time-varying** heat corrections, with one value (or a horizontal or three-dimensional field of values) per update time; the update frequency is controlled by the ndtupdfcorr namelist option, in units of model timesteps.

For a description of how to configure the time variable for time-varying bottom-layer temperatures, see §5.5 and the example contained therein.

Heat corrections may be applied at the surface only (L_FCORR) or at all model levels independently (L_FCORR_WITHZ). In the surface-only option, the intent is to correct the entire model-diagnosed mixed-layer depth based on the model's temperature bias at the surface. This type of correction is the same as

that calculated and applied via the L_RELAX namelist option, in which the model SSTs are relaxed towards a user-specified value with a user-specified relaxation timescale. Enabling L_RELAX allows the model to output a value fcorr at each horizontal gridpoint; this fcorr can then be used as an input to the model via the ancillary file described here. This allows the user to take means of the corrections—either over time, across an ensemble of simulations or both—and then use the mean correction to restore the model's temperatures towards climatology. Note, however, that in this method no attempt is made to correct the temperature of the water below the model-calculated mixed-layer depth, nor is any account taken of potential vertical variations in the temperature bias within the mixed layer.

In the depth-varying method, the intent is to correct the bias at each layer in the vertical independently. The model is not able to calculate these corrections itself, so they must be computed off-line. The preferred technique is to run KPP in forced mode and allow the model to drift for a desired period of time (e.g., one month). Once the "drifting" simulation is complete, compute the time-mean bias between the model temperatures and some observed temperatures throughout the depth of the model vertical domain; you may need to interpolate in the vertical grids of the model and the observations. From this drift, compute the amount of heat that must be added or removed from the model—at each three-dimensional grid point—to restore the model temperatures to the observations. To compute a monthly mean correction, the relevant equation is

```
DO i=1,nz
```

$$f(i) = \rho(i) * c_p(i) * \frac{(T_{obs} - T_{kpp})}{86400. * 15.} \tag{3}$$

```
ENDDO
```

where ρ is density, c_p is specific heat, T_{obs} is the observed monthly mean temperature and T_{kpp} is the KPP monthly mean temperature that has been allowed to drift.

Once these corrections have been calculated, the forced integration can be re-run with the corrections and (if desired) a second month can be run uncorrected. It is necessary to re-run the first month of the simulation to ensure a corrected mean state at the beginning of the second month. This procedure is more time-intensive than for the surface-only corrections, but it corrects all model levels—not just those within the mixed layer—and does not assume that the temperature is homogeneous with depth.

The composition of the netCDF ancillary file is similar for the two methods, with the difference being that the L_FCORR_WITHZ option requires that the ancillary file have a z dimension to allow the heat corrections to vary with depth.

An example of a depth-varying netCDF file is given as heat_corrections_with_depth.nc in the ancillaries directory.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to NX in parameter.inc
latitude	Latitude	Equal to NY in parameter.inc
z	Depth	Equal to NZP1 in parameter.inc
time	Time (only if L_UPDCLIM)	Equal to $\frac{\texttt{ndtocn}*86400*(\texttt{finalt}-\texttt{startt}+1)}{\texttt{dtsec}*\texttt{ndtupdfcorr}}$

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convection
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
z(z)	Depth (only if `L_FCORR_WITHZ`)	m	Negative down
time(time)	Time (only if `L_UPDCLIM`)	Days	Always positive
fcorr(longitude,latitude,z,time)	Heat corrections	W m^{-2} (`L_FCORR`) or W m^{-3} (`L_FCORR_WITHZ`)	Positive warms ocean

5.9 Surface forcing data

A surface forcing ancillary file is required if `L_FLUXDATA` is set to `.TRUE.`. The model will attempt to read the surface forcing data from the netCDF file specified in the `forcing_file` namelist option (§4.1). Additionally, the model will attempt to update the forcing data every `dtsec` seconds, or `ndtocn` timesteps. If you wish to use time-invariant forcing data, then you should set `dtsec` to the total number of seconds in the integration and `ndtocn` to the total number of timesteps you require.

For a description of how to configure the `time` variable for the time-varying surface forcing data, see §5.5 and the example contained therein.

The 3D KPP model requires the following surface fields at every horizontal gridpoint: net solar radiation, net longwave radiation, sensible heat flux, latent heat flux, precipitation minus evaporation and surface and meridional surface windstress. Sign conventions are particularly important for this ancillary file; they are described in the tables below.

An example of a time-varying surface forcing ancillary file (with daily mean values) is provided as `may_fluxes_daily_means.nc` in the `ancillaries` directory.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`
time	Time	Equal to $\frac{\texttt{dtocn}}{86400} * (\texttt{finalt} - \texttt{startt} + 1)$

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convention
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
time(time)	Time	Days	Always positive
swf(longitude,latitude,time)	Net short wave radiation	W m^{-2}	Positive from atmosphere to ocean
lwf(longitude,latitude,time)	Net long wave radiation	W m^{-2}	Positive from atmosphere to ocean
shf(longitude,latitude,time)	Sensible heat flux	W m^{-2}	Positive from atmosphere to ocean
lhf(longitude,latitude,time)	Latent heat flux	W m^{-2}	Positive from atmosphere to ocean
precip(longitude,latitude,time)	Precipitation minus evaporation	mm s^{-1}	Positive freshens ocean
taux(longitude,latitude,time)	Zonal surface windstress	kg m^{-2}	Positive westerly
tauy(longitude,latitude,time)	Meridional surface windstress	kg m^{-2}	Positive southerly

5.10 Jerlov water types

The Jerlov water type controls the radiative properties of the sea water. Five different types are available: I, IA, IB, II and III. The effect that each of these has on the opacity of the water to shortwave radiation can be seen in the `swfrac` subroutine in `ocn.f`.

An ancillary file of Jerlov water types is required if the `L_JERLOV` namelist option is set to `.TRUE.`. If this option is set to `.FALSE.` then all points are assumed to have a water type of IB. The model will attempt to read the Jerlov water types from the netCDF ancillary file specified in the `paras_file` namelist option. Note that there is currently no facility to read time-varying Jerlov water types.

In the ancillary file, a floating-point value should be specified at each horizontal gridpoint. The following table gives the correspondence between the value in the ancillary file and the Jerlov water type.

Value in ancillary	Jerlov water type
1	I
2	IA
3	IB
4	II
5	III

An example of this ancillary file is provided as `default_water_types.nc` in the `ancillaries` directory.

The netCDF file must contain the following dimensions:

netCDF name	Description	Length
longitude	Longitude	Equal to `NX` in `parameter.inc`
latitude	Latitude	Equal to `NY` in `parameter.inc`

The netCDF file must contain the following variables in the following dimensions:

netCDF name	Description	Units	Sign convention
longitude(longitude)	Longitude	Degrees	Positive east
latitude(latitude)	Latitude	Degrees	Positive north
jerlov(longitude,latitude)	Value corresponding to Jerlov water type	None	Always positive

6 Model output files

The 3D KPP model produces two types of output files: instantaneous and mean. The presence of these files is controlled by the `L_OUTPUT_INST` and `L_OUTPUT_MEAN` namelist options, respectively, with the output frequency controlled by the `ndtout` and `ndtout_mean` options, respectively.

All output from KPP is in netCDF format. By default, the model creates a new output file for every five days of the integration; this is done to prevent the files from becoming too large when using a high number of vertical points and requesting many variables. If the integration is less than five days long, then all of the output will be in a single file.

The following subsections list the available three-dimensional (i.e., lon x lat x depth) and single-level (i.e., lon x lat) fields that can be output from the model. If you wish to output fields that are currently not available, you can do so by adding your own code to `ncdf_out.f`. The code that is already contained in that file should show you the pattern you need to follow. After you add this code, you will also need to adjust the constants `N_VAROUTS` and `N_SINGOUTS` in `output.com` to match the new numbers of three-dimensional and single-level variables available, respectively.

Further subsections give details on the configuration of the instantaneous and mean output files.

6.1 Available three-dimensional variables

The following table lists the available three-dimensional (i.e., lon x lat x depth) fields that can be output from the 3D KPP model. Please note that the *order* in which the fields are listed is important. This order is the same as the order of the logical flags in the `L_VAROUT` and `L_MEAN_VAROUT` namelist options (§4.1). Thus, to enable or disable a particular output field, you must change the flag in the appropriate position in the list in those namelist options. For example, to disable output of temperature, change the third logical flag to `.F.`.

The table also lists the KPP variable in which the data are stored, the netCDF variable name (also called the "short name") in the output file, as well as the units in which the variable is output.

Position	KPP variable	netCDF	Units	Description
1	`U(:,:,1)`	u	m s^{-1}	Zonal velocity
2	`U(:,:,2)`	v	m s^{-1}	Meridional velocity
3	`X(:,:,1)`	T	°C	Temperature
4	`X(:,:,2)`	S	‰	Salinity
5	`buoy`	B	m s^{-2}	Buoyancy
6	`wU(:,0:NZP1-1,1)`	wu	m^2 s^{-2}	Turbulent zonal velocity flux
7	`wU(:,0:NZP1-1,2)`	wv	m^2 s^{-2}	Turbulent meridional velocity flux
8	`wX(:,0:NZP1-1,1)`	wT	°C m s^{-1}	Turbulent temperature flux
9	`wX(:,0:NZP1-1,2)`	wS	‰ m s^{-1}	Turbulent salinity flux
10	`wX(:,0:NZP1-1,3)`	wB	m^2 s^{-3}	Turbulent buoyancy flux
11	`wXNT(:,0:NZP1-1,1)`	wTnT	°C m s^{-1}	Non-turbulent temperature flux
12	`difm`	difm	m^2 s^{-1}	Diffusion coefficient for momentum
13	`dift`	dift	m^2 s^{-1}	Diffusion coefficient for temperature
14	`difs`	difs	m^2 s^{-1}	Diffusion coefficient for salinity
15	`rho`	rho	kg m^{-3}	Density
16	`cp`	cp	J kg^{-1} $°C^{-1}$	Specific heat capacity

6.2 Available single-level variables

The following table lists the available single-level (i.e., lon x lat) fields that can be output from the 3D KPP model. As for the three-dimensional variables, the *order* in which the fields are listed is important. This order is the same as the order of the logical flags in the `L_SINGOUT` and `L_MEAN_SINGOUT` namelist options (§4.1). Thus, to enable or disable a particular output field, you must change the flag in the appropriate position in the list in those namelist options. For example, to enable the output of the coupling weight, change the fifth logical flag to `.T.`.

The table also lists the KPP variable in which the data are stored, the netCDF variable name (also called the "short name") in the output file, as well as the units in which the variable is output.

Position	KPP variable	netCDF	Units	Description
1	`hmix`	hmix	m	Model-diagnosed mixed-layer depth
2	`fcorr`	fcorr	W^{-2}	Heat correction necessary to correct all of the water down to the model-diagnosed mixed-layer depth by the bias in the model SST, using the specified relaxation timescale. This is the output generated by enabling `L_RELAX` and giving a relaxation timescale `relax_in`. You may then use this output as input for the `L_FCORR` option.
3	`sflux(:,3,5,0)`	solar_in	W m^{-2}	The solar radiation received from the coupler. Applies only when the model runs in coupled mode; in forced mode, this will be zero.
4	`sflux(:,4,5,0)`	nsolar_in	W m^{-2}	The non-solar radiation received from the coupler. Applies only when the model runs in coupled mode; in forced mode, this will be zero.
5	`cplwght`	cplwght	None	The coupling weight used in the model. Applies only when a coupling weight is specified via the `L_CPLWGHT` namelist option (§4.1).

6.3 Instantaneous output files

The instantaneous netCDF output files are named `KPPocean_[last_time].nc`, where `last_output_time` is the last model timestep (in days) before the output file was closed. By default, each output file contains up to five days of data; therefore, `last_time` is always a multiple of five. Thus if you specify a `startt` of 0 in the namelist, the first output file will be called `KPPocean_0005.nc`, the second will be called `KPPocean_0010.nc` and so on.

Each output file will contain the three-dimensional and single-level fields that you specified in the `L_VAROUT` and `L_SINGOUT` namelist options. The file also contains the following dimensions and corresponding variables:

Dimension and variable name	Description
longitude	The longitude coordinate axis, in units of degrees (positive east)
latitude	The latitude coordinate axis, in units of degrees (positive north)
z	The depth coordinate axis (the midpoint of each layer in the vertical), in units of metres (negative down)
d	The depth coordinate axis (the boundaries of the layers in the vertical), in units of metres (positive down)
h	The depth of each layer in the vertical
time	The time coorindate axis, with one value per output time. The output frequency is controlled by the ndtout namelist option.

6.4 Mean output files

The mean netCDF output files are named KPPocean_[last_time]_means.nc, where last_time is the last model timestep (in days) before the output file was closed. By default, each output file contains up to five days of data; therefore last_time is always a multiple of five. Thus if you specify a startt of 3 in the namelist, the first output file will be called KPPocean_0008.nc, the second will be called KPPocean_0013.nc and so on.

Each output file will contain the three-dimension and single-level fields that you specified in the L_MEAN_VAROUT and L_MEAN_SINGOUT namelist options. The file also contains the dimensions and variables specified for the instantaneous output files above (§6.3).

7 Restarting an integration

To restart an integration of the 3D KPP model, you must have a restart file that was written at the end of the integration you wish to resume. The L_RESTARTW namelist option (§4.1) controls whether the model writes a restart file. It is not currently possible to write restart files at any time other than at the end of the integration. The name of the restart file is controlled by the restart_outfile namelist option.

Restart files are in an internal format; they are not netCDF files. You should not need to modify them or analyze them. They are intended purely to be read into and written out of the model.

Conducting a restart integration requires setting the following namelist options:

```
L_RESTART = .TRUE.
restart_infile = [name of restart file]
startt = [ending time (finalt) of last integration]
```

The last line is particularly important, as the value of startt must match the time that was written to the restart file at the end of the previous integration. This time is equal to the value of finalt that was used in the previous integration.

8 Revision History

8.1 Revisions of the 3D KPP model

The following are the changes made in each public release of the 3D KPP model:

- Release 2 – ?? July 2009
  - Preliminary support for coupling to the GFS atmospheric model, via code in `couple_io_cfs.f` and the `cpp` option `-DCFS`, to be used in conjunction with the existing option `-DCOUPLE`. Includes output of SSTs in GRIB format.
  - Added support for the IBM `xlf` compiler, including a new `cpp` option `-DOLD_NAME`.
  - New namelist option `ndt_per_file`, which controls the number of ocean timesteps between creating new output files. This option controls both the instantaneous and mean output files.
  - The `parameter.inc` and `landsea.com` files are pre-processed by `cpp` before compilation. The former is necessary to set the default grid for the GFS-coupled version (spectral T62 grid), as opposed to the grid for the forced and OASIS-coupled versions (regular N144 grid). The latter is necessary to add a variable for reading in the global land-sea mask, which is required for the GFS-coupled version.
- Release 1 – 13 May 2009
  - First public release of the code

8.2 Revisions to these instructions

The following are the changes made to these instructions, which accompany each public release of the 3D KPP model:

- Release 2 – ?? July 2009
  - Clarified our position on the use of this model in published research.
  - Removed `3D_ocn.nml` as standard input for the five-day test case, as this caused errors on the NCAR Bluefire machine. The model opens the namelist on unit 75 anyway, so supplying it on standard input as well was unnecessary.
  - Added a description for the `ndt_per_file` namelist option.
  - Noted that the `L_OUTKELVIN` namelist option is enabled when coupled to the GFS, as well as when coupled to OASIS.
- Release 1 – 13 May 2009
  - First public release of these instructions

References

Klingaman, N. P., H. Weller, S. J. Woolnough, P. M. Inness, and J. M. Slingo, 2009: Coupled simulations of the indian monsoon intraseasonal oscillation using a fine-resolution mixed-layer ocean model. *Proc. ECMWF Workshop on Ocean-Atmosphere Interaction (2008)*, European Centre for Medium-Range Weather Forecasts, Reading, England, 195–2005.

Large, W., J. McWilliams, and S. Doney, 1994: Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Rev. Geophys.*, **32**, 363–403.

Woolnough, S. J., F. Vitart, and M. A. Balmaseda, 2007: The role of the ocean in the Madden–Julian Oscillation: Implications for MJO prediction. *Quart. J. R. Meteorol. Soc.*, **133**, 117–128.